applied sciences

MDPI

Article

Class Imbalance in IoMT Datasets: Evaluating Balancing Strategies for Learning-Based Attack Detection

Eren Gencturk [1], Beste Ustubioglu [1,*], Guzin Ulutas [1] and Iraklis Symeonidis [2]

[1] Department of Computer Engineering, Karadeniz Technical University, Trabzon 61080, Turkey; ktu@erengencturk.com.tr (E.G.); gulutas@ktu.edu.tr (G.U.)
[2] RISE—Research Institutes of Sweden, Isafjordsgatan 22, 164 40 Stockholm, Sweden; iraklis.symeonidis@ri.se
* Correspondence: bustubioglu@ktu.edu.tr

Abstract

Internet of Medical Things (IoMT) devices are inherently vulnerable to cyberattacks, typically due to their limited processing power and memory capacity. Their widespread use in healthcare poses a significant security risk, threatening patient data privacy and the continuity of services. This study examines the effects of data imbalance correction and balancing strategies on the performance of machine and deep learning models using openly available IoMT datasets. In this context, four different balancing methods—RandomUnderSampler, SMOTE, Borderline-SMOTE, and ADASYN—were applied to three open-access IoMT datasets: ECU-IoHT, WUSTL, and CICIoMT2024. Performance analyses were conducted using five machine learning algorithms (AdaBoost, Logistic Regression, Random Forest, XGBoost, and K-Nearest Neighbor (KNN)) and two deep learning algorithms (Convolutional Neural Networks (CNN) and Deep Neural Networks (DNN)). In the highly imbalanced binary setting of the CICIoMT2024 dataset, the combination of RandomUnderSampler and SMOTE under the balanced-training/original-testing scenario produced the strongest improvement in the binary CICIoMT2024 setting, increasing the F1-Score from the unbalanced baseline to 99.87% for Random Forest and 99.86% for XGBoost across repeated runs. However, the benefit of balancing was not universal. In datasets with stronger class separability, such as ECU-IoHT, and in several multi-class settings, the effect of balancing was limited or, in some cases, inferior to the unbalanced baseline. These findings indicate that balancing is most effective under specific conditions, particularly in highly imbalanced binary tasks, and should be validated using class-sensitive metrics rather than overall performance alone.

Keywords: Internet of Medical Things (IoMT); data balancing; machine learning; deep learning; cybersecurity; healthcare

Check for updates

Academic Editor: Paolino Di Felice

Received: 5 March 2026
Revised: 30 March 2026
Accepted: 2 April 2026
Published: 15 May 2026

Copyright: © 2026 by the authors. Licensee MDPI, Basel, Switzerland. This article is an open access article distributed under the terms and conditions of the Creative Commons Attribution (CC BY) license.

1. Introduction

The Internet of Things (IoT) enables physical devices, vehicles, home appliances, and other objects to communicate over the internet by equipping them with sensors, software, and connectivity features [1]. This technology allows devices to collect and analyze data and to make intelligent decisions based on this information [2]. The main components of IoT include devices and sensors, connectivity technologies, data processing systems, and user interfaces [3]. Communication within the IoT ecosystem relies on protocols designed for different needs, such as MQTT for lightweight real-time data transmission, CoAP for resource-constrained devices, and LoRaWAN for long-range low-power applications [4–6].

Additionally, connectivity technologies such as Wi-Fi, Bluetooth, and Zigbee play a significant role in enabling devices to communicate with each other and with cloud systems [7].

IoT technologies are used across many sectors. In healthcare in particular, IoT-based wearable devices allow for continuous monitoring of patients' health status, making services more efficient and personalized [8]. Recent reports estimated that approximately 18.8 billion IoT devices were connected worldwide, with projections indicating that this number would reach 30.9 billion by the end of 2025 [9]. Specifically in the healthcare sector, IoT devices were projected to approach 1 billion by 2025, largely due to the increasing demand for wearable health devices and remote patient monitoring systems [9]. Likewise, the global IoT market size was projected to exceed 1 trillion dollars by 2025, driven by rapidly expanding application areas and increasing investments [7].

IoMT devices support several critical functions in healthcare, including continuous patient monitoring, smart drug delivery through automated infusion systems, and telemedicine services that allow remote diagnosis and treatment [8]. These capabilities improve treatment effectiveness and patient comfort, while also reducing the need for physical check-ups. However, the widespread deployment of such devices also enlarges the cyberattack surface, particularly because many IoMT devices operate under strict resource and security constraints [10].

Attack types targeting IoMT infrastructures include a wide variety of methods such as Smurf Attacks, ARP Spoofing, MQTT-based DoS/DDoS attacks, and reconnaissance techniques [11]. For instance, Connect Flood and Publish Flood attacks via MQTT consume system resources and render devices non-functional [12], while reconnaissance techniques such as port scanning and vulnerability scanning lay the groundwork for more targeted future attacks [13]. These attacks can compromise patient data privacy, disrupt critical healthcare services, and undermine data integrity, making robust intrusion detection essential for IoMT environments.

Machine learning and deep learning methods have been increasingly adopted for IoMT intrusion detection due to their ability to identify complex attack patterns in network traffic [10]. However, a persistent challenge in this domain is class imbalance: attack traffic, especially from rare attack types, typically constitutes only a small fraction of network data. Models trained on such imbalanced datasets tend to favor the majority class and miss minority attack patterns, which has direct security implications in healthcare environments.

In IoMT intrusion detection, class imbalance is not merely a statistical issue but also a security-critical problem. Rare attack types are often underrepresented in training data, causing models to produce false negatives for minority classes. In operational healthcare environments, such missed detections mean that malicious activities may remain unnoticed until they compromise patient privacy, data integrity, or service availability. From this perspective, data balancing contributes to security by improving model sensitivity to underrepresented attack classes and supporting more reliable detection of diverse attack behaviors. Therefore, evaluating balancing strategies is important not only for predictive performance but also for strengthening the practical security value of learning-based IoMT intrusion detection systems.

This study investigates how balancing strategies affect the performance of machine learning and deep learning models on three open-access IoMT datasets: ECU-IoHT, WUSTL, and CICIoMT2024. Four balancing methods—RandomUnderSampler (RUS), SMOTE, Borderline-SMOTE, and ADASYN—are evaluated across five classical machine learning models (AdaBoost, Logistic Regression, Random Forest, XGBoost, KNN) and two deep learning models (CNN, DNN). The goal is to identify how balancing interacts with model family, task type, and dataset characteristics in IoMT intrusion detection.

The main contributions of this study are as follows. First, we provide a cross-dataset comparison of balancing strategies over three complementary IoMT scenarios: ECU-IoHT represents a packet-level attack-detection setting with relatively strong class separability; WUSTL represents a healthcare-oriented intrusion-detection setting with binary attack discrimination; and CICIoMT2024 represents a large-scale and highly imbalanced benchmark with both binary and multi-class attack structures. Second, rather than proposing a new balancing algorithm, we systematically evaluate foundational and widely used balancing strategies and their combinations under a unified benchmark design, allowing their effects to be compared across model families, task types, and dataset characteristics. Third, we analyze not only overall predictive performance but also class-sensitive behavior through metrics such as PR-AUC, FNR, minority-class recall, and cost-oriented missed-detection summaries, thereby strengthening the security interpretation of balancing in IoMT intrusion detection. Finally, we translate the empirical findings into practical guidance by relating balancing effectiveness to imbalance severity, task type, and model family.

In the following sections of the paper, Section 2 presents a literature review on IoMT security, methods for handling imbalanced datasets, and machine/deep learning-based attack detection approaches. Next, Section 3 Materials and Methods provides a detailed explanation of the IoMT datasets used in the study, the balancing strategies applied, and the selected machine and deep learning models. Performance evaluation metrics, experimental results, and analysis of these results are presented in detail under Section 4 Results and Discussion. Finally, Section 5 summarizes the main findings of the study and offers recommendations for future work.

2. Related Work

Today, the security of IoMT devices has become increasingly important due to the growing number of devices in use and the sensitivity of the data they handle. In this section, a literature review is presented on IoMT security, methods for handling imbalanced datasets, and machine/deep learning-based attack detection approaches. In particular, the effects of imbalanced data distributions on model performance, data balancing techniques used to address this issue, and machine and deep learning-based solutions developed for threat detection in IoMT environments are discussed.

Sridevi et al. (2022) compared the performance of various machine learning algorithms for the detection of attacks in network traffic and proposed a combined classification method called the Voting Classifier, which incorporates Support Vector Machines (SVM), K-Nearest Neighbors (KNN), Random Forest, and Voting Classifier itself [14]. The study used the NSL-KDD dataset, a modified version of the KDD99 dataset, which includes 42 features and 4 types of attacks. The overall accuracy rates reported in the study were 98.56% for SVM, 98.8% for KNN, 99.19% for Random Forest, and 99.41% for their own Voting Classifier model. The study reported that the Voting Classifier, which combines SVM, KNN, and Random Forest algorithms, produced superior results in both accuracy and F1-Score compared to using each algorithm individually.

Arshed et al. (2022) compared the performance of various machine learning algorithms for detecting attacks and anomalies in IoT environments and specifically evaluated the impact of data balancing techniques on imbalanced data [15]. The open-source dataset used in the research was divided into 8 classes and included 13 features. The authors emphasized that imbalanced datasets cause models to overfit, with majority class samples being learned predominantly while minority classes are neglected. To address this issue, they used the RandomUnderSampler method to create a balanced dataset, resulting in a fairer model comparison. Experimental results showed that the Generalized Linear Model algorithm achieved better results than other algorithms on both balanced (94.3% accuracy)

and imbalanced (98% accuracy) datasets. Random Forest and Naive Bayes algorithms also demonstrated high performance.

In a study by Zhang and Liu (2022), a new data augmentation based attack detection method was proposed to detect especially minority classes in IoT network attacks [16]. This method, called ICVAE-BSM, combines an Improved Conditional Variational Autoencoder (ICVAE) with the Borderline-SMOTE (BSM) algorithm. The proposed model generates high-representational synthetic samples for attack types with few samples, thereby reducing data imbalance and increasing overall classification performance. The study used three different datasets (NSL-KDD, CIC-IDS2017, and CSE-CIC-IDS2018) each covering different types of IoT network attacks. The model achieved significantly improved performance, especially for minority attack classes such as R2L, U2R, Web Attack, and Infiltration. In terms of performance evaluation, the ICVAE-BSM model outperformed classical data augmentation methods such as GAN-DNN, CVAE-DNN, and RF-SMOTE in F1-Score, AUC, AUC-PR, and accuracy. For example, it achieved a G-mean of 94.3% and accuracy of 86.5% on the NSL-KDD dataset, and an F1-Score of 90.2% on the IDS2017 set.

In another study by Farhan and Jasim (2022), a Long Short-Term Memory (LSTM) architecture was used to develop an attack detection system for network traffic cyberattacks [17]. The CSE-CIC-IDS2018 dataset used in the study is based on real-time traffic data collected from Amazon Web Services (AWS) infrastructure and contains more than 1.25 million samples and 10 different types of attacks (e.g., DDoS-HOIC, DDoS-LOIC, Botnet, FTP-BruteForce, SSH-BruteForce). The proposed LSTM model consists of three layers; ReLU activation was used in the first two layers and Softmax in the final layer. According to the experimental results, the model achieved 99% accuracy.

Dey and Pratap (2023) compared three different data augmentation techniques (SMOTE, Borderline-SMOTE, and ADASYN) to improve classification performance on imbalanced datasets [18]. The datasets used included Pima Indian Diabetes, Breast Cancer Wisconsin Diagnostic, and Insurance Claim Prediction, all of which have imbalanced class distributions. The number of samples in minority and majority classes differed in each dataset, which can negatively impact the predictive performance of machine learning models. The performance of the sampling techniques was evaluated using SVM, KNN, Gaussian Naive Bayes, Decision Trees, and Random Forest classifiers. Comparisons were made based on accuracy, precision, recall, F1-Score, and AUC (Area Under the ROC Curve) metrics. According to the results: the Random Forest classifier generally achieved the highest accuracy and AUC values across all sampling techniques and datasets. SMOTE achieved up to 99.3% accuracy on the Breast Cancer dataset, and when used with Random Forest, achieved 81.5% accuracy on the Insurance Claim dataset. Borderline-SMOTE, particularly when used with SVM and Decision Tree, produced better results on the Pima Indian and Breast Cancer datasets. ADASYN, targeting hard-to-learn minority samples, provided better results on some datasets, achieving 99% accuracy with Decision Tree on the Insurance Claim dataset. The study showed that different data augmentation techniques achieved varying successes with different classifiers and datasets.

Dadkhah et al. (2024) introduced a new open dataset, CICIoMT2024, by creating a comprehensive and realistic test environment to evaluate the security of IoMT devices [19]. The study aimed to address the shortcomings of existing datasets, such as limited device variety, lack of attack types, and lack of device profiling support. A total of 18 different types of attacks were carried out using 40 IoMT devices (25 real, 15 simulated) supporting different protocols (Wi-Fi, MQTT, Bluetooth), and all network traffic was collected in detail. The attacks were classified into five main categories: DDoS, DoS, Recon, MQTT-based attacks, and Spoofing. The collected data included not only attack behaviors but also comprehensive profiling data reflecting device lifecycle (joining the network, idling, active

use, interaction). This provides a rich resource for both traditional attack detection and anomaly-based security solutions.

The CICIoMT2024 dataset stands out not only for its new device varieties and attack types, but also for its high-quality network monitoring equipment (e.g., Faraday cage, network tap systems) and detailed feature engineering. The collected data is provided in PCAP and CSV formats and has been tested with methods such as Logistic Regression, AdaBoost, Random Forest, and Deep Neural Network (DNN). In three classification scenarios (binary, categorical, and multi-class), Random Forest and DNN algorithms performed best, especially in the multi-class setting (19 classes), achieving up to 73% accuracy and 55% F1-Score.

Unlike previous studies that often focus on a single dataset, a limited set of balancing methods, or a single model family, our study is designed as a controlled comparative benchmark across three complementary IoMT scenarios. The goal is not to claim a new balancing algorithm, but to clarify when foundational balancing strategies remain effective, when their benefits become limited, and how their behavior changes across binary versus multi-class tasks and across classical versus deep learning models. In this sense, the contribution of the study lies in the unified cross-dataset evaluation framework, the explicit comparison of representative balancing philosophies, and the security-oriented interpretation of class-sensitive detection behavior.

The evaluation of the current literature reveals that there are still areas where further contributions can be made to IoMT security and that there is a basis for a more comprehensive understanding of the subject. In this context, the following section presents detailed information about the datasets used in this study, the applied balancing techniques, and the performance evaluation metrics.

The literature on data balancing has evolved significantly in recent years, moving beyond the foundational resampling techniques evaluated in this study. Modern approaches can be broadly categorized into advanced data-level resampling, generative modeling, and algorithm-level balancing. Advanced resampling techniques aim to improve upon SMOTE's core mechanism. For instance, cluster-aware methods like KMeans-SMOTE first partition the feature space to generate samples only in "safe" clusters, while hybrid methods such as SMOTE-ENN or SMOTE-TomekLinks couple synthetic oversampling with data cleaning to remove noisy samples near class boundaries [20,21]. Generative models represent the state-of-the-art for creating high-fidelity synthetic data. While our study acknowledges early generative approaches like GANs, recent advancements for tabular data include CTAB-GAN+ and diffusion-based models like TabDDPM, which often yield superior synthetic samples and can serve as powerful minority oversamplers [22,23]. Although promising, these methods are computationally intensive and highly sensitive to hyperparameter tuning for high-dimensional network flow data. Lastly, algorithm-level balancing offers an alternative paradigm that modifies the learning process itself, rather than the data distribution. Techniques such as Focal Loss, Class-Balanced Loss (which re-weights the loss based on the effective number of samples), and logit adjustment directly address class imbalance during model training, proving especially effective for deep learning models [24,25]. More broadly, recent studies in privacy-preserving federated learning, transfer-based diagnosis, and few-shot cross-domain learning also highlight the importance of robustness, distribution shift awareness, and data efficiency in safety-critical intelligent systems. Although these studies are not direct balancing baselines for IoMT intrusion detection, they reinforce the broader motivation for evaluating how data distribution and learning strategy affect reliable detection behavior in practice [26–28].

3. Materials and Methods

In our study, balancing procedures were performed on open-source IoMT datasets—ECU-IoHT, WUSTL, and CICIoMT2024—to examine the effects on the performance of machine learning and deep learning methods. In total, five different machine learning and two different deep learning methods were applied to three different open-source IoMT datasets. The workflow of the method is shown in Figure 1.

The balancing procedures were applied to the samples designated as training data; RandomUnderSampler was used in combination with ADASYN, Borderline-SMOTE, and SMOTE methods, each applied separately. For classes with a high number of samples, undersampling was applied, while for classes with fewer samples, oversampling was performed. The amount of increase or decrease was determined based on the average number obtained by dividing the total number of samples by the number of classes.

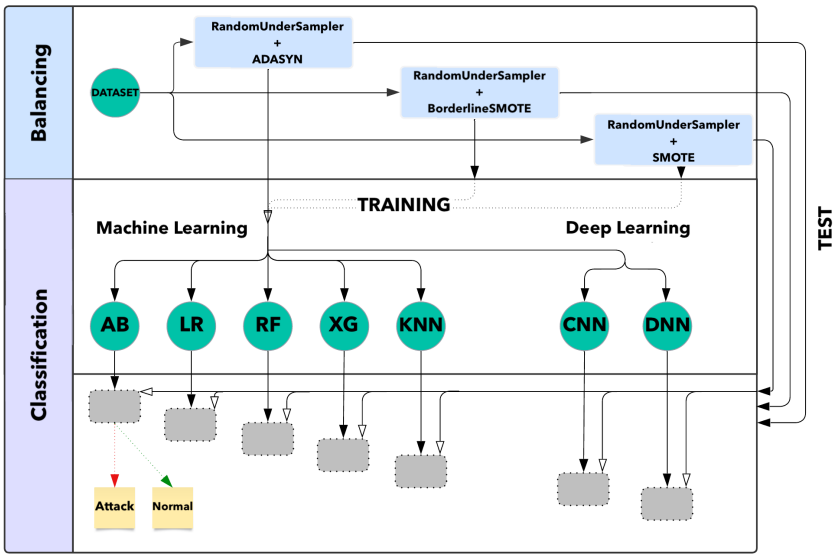


Figure 1. Flowchart of the proposed method.

Three datasets are used: one provides only binary labels (attack vs. normal), while the other two include both binary and multi-class annotations (attack types and category labels). For CICIoMT2024, the predefined 80%/20% train–test split is adopted; the remaining two datasets are split 80%/20% by us. In addition to results on the original, imbalanced test sets, we also report a balanced-test variant obtained by downsampling each class to the size of the minority class. Detailed dataset descriptions and procedures are provided in the next section.

Balancing was applied only to the training data. The original test set was retained as the primary evaluation setting because it preserves the naturally occurring class distribution and therefore better reflects deployment-oriented IoMT intrusion-detection conditions. In addition, a balanced-test variant was constructed only for diagnostic analysis, in order to examine whether strong overall scores under imbalanced testing were masking weaknesses on minority classes. Accordingly, results on the original test distribution are treated as the main benchmark, whereas the balanced-test results are reported as a supplementary class-sensitive evaluation rather than as the primary operational scenario. Therefore, the balanced-test setting should not be interpreted as a deployment-oriented benchmark, but as a controlled diagnostic stress test used to reveal class-sensitive weaknesses that may remain hidden under the original test distribution.

3.1. Datasets

The dataset is one of the most critical elements forming the foundation of machine learning and artificial intelligence applications. These datasets are shaped by information obtained from observations, measurements, and experiences related to a specific problem, providing an indispensable basis for training, testing, and evaluating the performance of learning algorithms [29]. In recent years, especially with the widespread use of artificial intelligence in fields such as finance, healthcare, and cybersecurity, the importance of datasets in terms of both volume and quality has significantly increased [30]. The performance of a model largely depends on the richness, accuracy, and representativeness of the data provided to it [31].

The main function of datasets is to provide a source that contains patterns and relationships which an AI model can learn and which reflects the real world. However, this function is not limited to collecting a large number of data samples; these data must also be diverse, covering different classes, variations, and conditions. For instance, datasets used in cybersecurity, such as NSL-KDD or CIC-IDS2018, aim to meet this need by including various types of attacks and network traffic behaviors [32,33]. The quality of a dataset is measured by key criteria such as representativeness, class balance, label accuracy, meaningful features, and compliance with privacy and ethical standards [34]. If there are imbalanced class distributions in a dataset (e.g., when there are far fewer attack samples compared to normal data), the model may have difficulty learning minority classes, which can negatively affect overall performance. Synthetic data generation techniques such as SMOTE or ADASYN are used to address such situations, making the dataset more balanced and functional [35].

The process of preparing a dataset may seem like a straightforward data collection task from the outside, but in reality, it is quite complex and multi-layered. The first step is to decide whether the data will be collected from the real world or simulation environments, followed by feature engineering to extract meaningful information from raw data [36]. The labeling stage requires correctly classifying the data, which can be time-consuming and error-prone, especially for large datasets. Next comes data cleaning, where missing, inconsistent, or noisy data is removed. Especially in sensitive areas such as IoT and healthcare, anonymization and security filters are crucial due to the presence of personal data. Compliance with ethical and legal regulations during this process can make sharing or preparing datasets more challenging [37].

Today, datasets have moved beyond being just a tool for training models; they also play a broader role in algorithm comparison, scientific reproducibility, standardization, and accuracy [38]. Open-access, well-documented, and regularly updated datasets have become a common language in the AI community. Such datasets make researchers' work sustainable and allow different models to be tested under the same conditions. For example, the recently released CICIoMT2024 dataset used in this study is a unique resource for both device profiling and attack detection, thanks to its multi-protocol traffic data collected from IoMT devices [19]. These developments show that the evolution of datasets is not only quantitative but also qualitative; datasets are now being specialized to meet the needs of specific fields.

In conclusion, the dataset stands out as the fundamental element that determines not only what an AI system has learned but also what it can learn. A comprehensive, balanced, accurately labeled dataset that reflects the real world directly shapes the performance of a model; otherwise, even the most advanced algorithms may perform poorly with inadequate data. Therefore, the process of dataset selection and preparation should be approached with as much care as model design in AI projects. In the future, the further development of field-specific datasets for healthcare, cybersecurity, or autonomous systems will enhance

both the accuracy and applicability of artificial intelligence. For this reason, the future of AI largely depends on the evolution of datasets and innovations in this area.

3.1.1. WUSTL EHMS 2020

The WUSTL EHMS 2020 Dataset was created using a real-time Enhanced Health Monitoring System (EHMS) test environment for the purpose of cybersecurity research in the Internet of Medical Things (IoMT) field [39]. This test environment collects both network flow metrics and patients' biometric data due to the lack of datasets that combine biometric and network data.

The EHMS test environment is divided into four parts: medical sensors, gateway, network, and visualization & control. The data flow starts from sensors attached to the patient's body and continues to the gateway. The gateway then sends the data to the main server for visualization via a switch and router. An attacker can intercept these data before they reach the server. The Intrusion Detection System (IDS) is responsible for capturing real-time network flow traffic and the patient's biometric data, as well as detecting anomalies.

The WUSTL dataset consists of a total of 16,318 samples (rows) and 44 features (columns). The number of samples belonging to attack and normal classes is presented in Table 1. The main purpose of the dataset is to analyze network traffic, especially in IoT and health-based systems, and distinguish between normal and attack-related activities. In this context, both technical network traffic measurements and patient data from medical devices are collected together.

The features in the dataset can be grouped into three main categories: network traffic features, patient sensor data, and label information. Traffic features include source and destination address information (SrcAddr, DstAddr, Sport, Dport), direction (Dir), flags (Flgs), and data volumes (SrcBytes, DstBytes, TotBytes). In addition, the dataset contains attributes that enable detailed traffic analysis, such as load (Load, SrcLoad, DstLoad), timing (Dur, SrcGap, DstGap, SIntPkt, DIntPkt), and loss rates (Loss, pLoss, pSrcLoss, pDstLoss).

The biometric data obtained from medical devices is a distinguishing feature of this dataset compared to similar ones. The measured values include body temperature (Temp), blood oxygen level (SpO2), pulse rate (Pulse_Rate), systolic and diastolic blood pressure (SYS, DIA), heart rate (Heart_rate), and respiratory rate (Resp_Rate).

Table 1. Sample counts in the WUSTL dataset.

Class	Count
Normal	14,272
Attack	2046

3.1.2. ECU-IoHT

To strengthen the cybersecurity of IoHT, the ECU-IoHT dataset was developed, built upon an IoHT environment where various attacks exploiting different vulnerabilities were carried out [40]. This dataset was designed to help the healthcare security community analyze attack behaviors and develop robust countermeasures. In this dataset, an IoHT test environment was set up, and vulnerabilities were investigated to develop a realistic dataset. The dataset features two classes (attack and normal) as well as five attack types: Smurf attack, Nmap Port Scan, ARP Spoofing, DoS attack, and normal.

As seen in Table 2, the dataset consists of a total of 111,206 samples, with each row representing a network packet or event and containing nine features: No, Time, Source, Destination, Protocol, Length, Info, Type, and Type of attack. The "No" column uniquely identifies each network event, while the "Time" column shows the timestamp of events

in seconds. The "Source" and "Destination" columns indicate the IP or MAC addresses from which the packets are sent and received. The "Protocol" column includes network protocols such as ARP, TCP, and DNS, while the "Length" column expresses the packet size in bytes. The "Info" column provides detailed information about the packet content, and the "Type" and "Type of attack" columns label the events as either normal or attack, and specify the attack types.

Table 2. Sample counts in the ECU-IoHT dataset.

Class	Type of Attack	Count
Attack	Smurf Attack	77,920
	Nmap Port Scan	6836
	ARP Spoofing	2359
	DoS Attack	639
Normal	-	23,453

The "Time" column shows that events occurred within the time interval from 0.0 s to 10,109 s. This data provides a valuable opportunity to examine the temporal distribution of attacks. For example, patterns such as the concentration of ARP Spoofing attacks in certain intervals or the increase of DoS attacks at specific times can be detected.

The "Length" column indicates the size of packets in bytes and is an important metric for network traffic analysis. ARP packets are generally 42 bytes, while TCP and TLSv1.1 packets have larger sizes such as 58 bytes, 321 bytes, or 395 bytes. These differences can be used to distinguish between normal and abnormal traffic patterns.

The "Info" column provides detailed information about packet contents. ARP queries, for example, "Who has 192.168.43.1? Tell 192.168.43.186", demonstrate device authentication. DNS queries such as "Standard query 0x0c44 PTR 1.43.168.192.in-addr.arpa" contain domain name resolution requests, while TCP connections are indicated with statements like "[SYN] Seq=0 Win=1024 Len=0 MSS=1460", describing the process of establishing a connection.

3.1.3. CICIoMT2024

The Internet of Things (IoT) is becoming increasingly integrated into daily life, particularly in the field of healthcare through the Internet of Medical Things (IoMT). IoMT devices enhance healthcare services by supporting functionalities such as uninterrupted operation. However, these devices also raise serious cybersecurity concerns due to vulnerabilities that may arise during health monitoring. This issue is further exacerbated by the complexity of IoMT network traffic and the large volume of data generated.

In this context, Machine Learning techniques are being used to detect, prevent, and mitigate the effects of cyberattacks. However, existing benchmark datasets have some fundamental shortcomings, such as the presence of only a small number of real devices, limited diversity, and the inability to create comprehensive device profiles. To address these shortcomings and to improve and evaluate IoMT security solutions, the CICIoMT2024 dataset has been proposed as a realistic benchmark dataset [19]. As part of this study, an IoMT test environment consisting of 40 devices (25 real, 15 simulated) was established.

The CICIoMT2024 dataset serves as a foundation that complements existing datasets and supports Machine Learning-based cybersecurity solutions. It provides a valuable resource for researchers to classify cyberattacks targeting IoMT devices and to develop secure healthcare systems, thus aiming to contribute to the widespread adoption of more reliable IoMT devices. The CICIoMT2024 training and test datasets are provided separately. The training set consists of 7,160,831 samples and the test set consists of 1,614,182 samples.

Class, category, and attack counts are presented in Table 3 for the training set and Table 4 for the test set. No additional train-test split was performed by us.

Table 3. Sample counts in the CICIoMT2024 training set.

Class	Category	Type of Attack	Count
Normal	-	-	192,732
Attack	Spoofing	ARP Spoofing	16,047
	DDoS	TCP_IP-DDoS-UDP	1,635,956
		TCP_IP-DDoS-ICMP	1,537,476
		TCP_IP-DDoS-TCP	804,465
		TCP_IP-DDoS-SYN	801,962
	DoS	TCP_IP-DoS-UDP	566,950
		TCP_IP-DoS-SYN	441,903
		TCP_IP-DoS-ICMP	416,292
		TCP_IP-DoS-TCP	380,384
	MQTT	MQTT-DDoS-Connect_Flood	173,036
		MQTT-DoS-Publish_Flood	44,376
		MQTT-DDoS-Publish_Flood	27,623
		MQTT-DoS-Connect_Flood	12,773
		MQTT-Malformed_Data	5130
	RECON	Recon-Port_Scan	83,981
		Recon-OS_Scan	16,832
		Recon-VulScan	2173
		Recon-Ping_Sweep	740

Table 4. Sample counts in the CICIoMT2024 test set.

Class	Category	Type of Attack	Count
Normal	-	-	37,607
Attack	Spoofing	ARP Spoofing	1744
	DDoS	TCP_IP-DDoS-UDP	362,070
		TCP_IP-DDoS-ICMP	349,699
		TCP_IP-DDoS-TCP	182,598
		TCP_IP-DDoS-SYN	172,397
	DoS	TCP_IP-DoS-UDP	137,553
		TCP_IP-DoS-SYN	98,595
		TCP_IP-DoS-ICMP	98,432
		TCP_IP-DoS-TCP	82,096
	MQTT	MQTT-DDoS-Connect_Flood	41,916
		MQTT-DoS-Publish_Flood	8505
		MQTT-DDoS-Publish_Flood	8416
		MQTT-DoS-Connect_Flood	3131
		MQTT-Malformed_Data	1747
	RECON	Recon-Port_Scan	22,622
		Recon-OS_Scan	3834
		Recon-VulScan	1034
		Recon-Ping_Sweep	186

The dataset contains traces of attacks conducted over three main communication protocols: Wi-Fi, Bluetooth Low Energy (BLE), and MQTT. Under each protocol, both normal and various attack scenarios were modeled. The 18 different attacks applied are classified under five categories: DDoS, DoS, Recon (reconnaissance attacks), MQTT-based attacks, and Spoofing. These attacks include ARP spoofing, port scanning, ping sweep, MQTT publish/connect flood, malformed packet transmissions, and various TCP/UDP/ICMP/SYN attacks. Thus, the dataset is suitable for both multiclass and binary classification studies.

The dataset shows the count or average of packets belonging to relevant protocols such as TCP, UDP, ICMP, ARP, DHCP, DNS, and HTTP. It contains information such as fin_flag_number, syn_flag_number, and rst_flag_number, which are useful for detecting DoS and port scanning (recon) attacks. It also includes statistics such as Min, Max, AVG, and Std for packet lengths, which can be used for network load analysis.

The features extracted from the dataset cover both technical details at the network level and time-based behaviors. The features include packet header lengths, Time-To-Live (TTL), flag information (ACK, SYN, FIN, etc.), average values for protocols such as IGMP, DNS, and HTTP, as well as statistical information regarding packet sizes (minimum, maximum, average, and standard deviation). In total, 45 features were defined, and these features were provided in PCAP and CSV formats.

Since the main objective of this study is to evaluate balancing strategies under different degrees of class skew, the imbalance severity of each dataset was quantified using the imbalance ratio (IR), defined as the ratio between the majority-class count and the count of the target class. Table 5 summarizes these ratios across the datasets and highlights the substantial differences in class skew, particularly for minority attack categories in ECU-IoHT and CICIoMT2024. This variation in imbalance severity provides an important basis for interpreting the comparative effectiveness of the balancing strategies examined in the following sections.

Table 5. Class imbalance ratios (IR) for all datasets. IR = $\text{count}_{\text{majority}}/\text{count}_{\text{class}}$; IR = 1.00 denotes the majority class.

Dataset	Class/Attack Type	Sample Count	IR
WUSTL (2-class)	Normal	14,272	1.00
	Attack	2046	6.98
ECU-IoHT (2-class)	Attack	87,754	1.00
	Normal	23,453	3.74
ECU-IoHT (5-class)	Smurf Attack	77,920	1.00
	Normal	23,453	3.32
	Nmap Port Scan	6836	11.40
	ARP Spoofing	2359	33.03
	DoS Attack	639	121.94
CICIoMT2024 (2-class)	Attack	8,544,674	1.00
	Benign	230,339	37.10
CICIoMT2024 (19-class) (representative rows)	TCP_IP-DDoS-UDP	1,998,026	1.00
	Benign	230,339	8.67
	Recon-Port_Scan	106,603	18.74
	ARP Spoofing	17,791	112.31
	MQTT-Malformed_Data	6877	290.54
	Recon-VulScan	3207	623.02
	Recon-Ping_Sweep	926	2157.70

3.2. Data Balancing

Data imbalance refers to situations in classification problems where some classes have significantly more samples than others. The performance of machine learning and deep learning models largely depends on the quality of the dataset used for training. However, one of the common situations encountered in real-world classification problems is the unequal distribution of samples between classes—this is called an imbalanced dataset. For example, in fields such as fraud detection, medical diagnosis, or production line fault detection, the event of interest often forms a minority class, representing only a small portion of the data. This imbalance can lead to a loss in the performance of machine and deep learning algorithms.

Models trained directly on imbalanced datasets tend to develop a strong bias toward the majority class, as they generally aim to maximize overall accuracy or minimize total

error. A classifier may even achieve high accuracy simply by assigning all samples to the majority class. This prevents the model from effectively learning the patterns and distinguishing features of the minority class. Furthermore, failure to address data imbalance is not just a technical error but also a risk with ethical and practical consequences, and since deep learning models are complex structures, the risk of overfitting is even greater with imbalanced data. Overfitting to the majority class can lead to high-variance and unreliable predictions for the minority class. As a result, the model may perform well on the majority class while failing to detect minority class samples, which are often critically important.

Due to this bias, standard metrics can be misleading when evaluating model performance on imbalanced datasets. In particular, accuracy may conceal the model's true performance, especially its effectiveness on the minority class. Therefore, when working with imbalanced datasets, focusing on metrics such as Precision, Recall, and F1-Score is much more informative and reliable. These metrics provide a more balanced view of how well the model distinguishes both positive and negative classes.

Data balancing methods aim to reduce this bias by helping the model learn the minority class more effectively. The main approaches include resampling techniques: oversampling increases the number of minority class samples or generates synthetic data, while undersampling reduces the number of majority class samples to balance the dataset. These techniques help the model focus equally on both classes, enabling it to produce more balanced and reliable predictions [19,41].

In IoMT cybersecurity, minority classes often correspond to less frequent but still security-relevant attack behaviors. If these classes are under-learned, an IDS may achieve high overall accuracy while failing to detect precisely the attacks that enable service disruption, traffic manipulation, reconnaissance, or unauthorized exposure of sensitive medical information. Therefore, the goal of balancing is not only to improve classification scores, but also to reduce false negatives for underrepresented attack categories and to enhance the robustness of security monitoring. In this sense, balancing should be viewed as a mechanism that supports the confidentiality, integrity, and availability objectives of IoMT systems through more dependable attack detection.

In this study, four different balancing methods were applied to the datasets to mitigate the negative effects of inter-class imbalance on model performance. These methods are explained below:

SMOTE (Synthetic Minority Over-sampling Technique): This is a method used to increase the number of minority class samples. In this technique, a certain number of nearest neighbors are determined for each minority class sample; then, synthetic samples are created at intermediate points calculated using the vector differences between each sample and its neighbors [35]. Thus, instead of simply duplicating existing samples, synthetic new samples are generated along the difference vectors between existing samples, which expands the dataset and improves class balance. However, SMOTE may lead to class overlap and overfitting, especially in boundary regions [42].

Borderline-SMOTE: This technique takes a more strategic sampling approach by dividing minority class samples into three categories based on their positions [43]. In this classification, "noise" refers to minority samples whose neighbors are all from the majority class—these are usually considered erroneous or noisy. "Safe" samples are those found in safer regions, mostly surrounded by other minority class samples. "Danger" samples are those whose neighbors are mostly from the majority class and thus are located near the class decision boundary. Borderline-SMOTE generates synthetic samples using only these "danger" samples, enabling the model to learn patterns near the decision boundary more effectively.

ADASYN (Adaptive Synthetic Sampling): Like SMOTE, ADASYN produces synthetic samples, but it does so in a more adaptive and data density-sensitive manner [44]. It determines the number of synthetic samples to generate for each minority sample according to how difficult it is to learn. Samples that are close to the decision boundary and more difficult to learn are represented with more synthetic data. In this way, ADASYN directs the model's attention to more complex and critical patterns. However, in some cases, the generation process may lead to class overlap, which can negatively affect classifier accuracy.

RandomUnderSampler: This method balances the dataset by randomly reducing the number of majority class samples in imbalanced datasets [20]. It is a simple, fast, and computationally inexpensive approach. However, random removal of samples may result in the loss of important patterns and informative samples. Especially if samples near class boundaries are removed, the model's decision mechanism may be disrupted. Therefore, this method should be used with caution and is often recommended in combination with other techniques.

In this study, four foundational and widely adopted methods—RandomUnderSampler, SMOTE, Borderline-SMOTE, and ADASYN—were selected to evaluate the impact of data balancing strategies. The rationale for this choice is threefold. First, our primary objective is to establish a comprehensive baseline by systematically comparing foundational balancing approaches. Their well-documented behavior ensures a high degree of interpretability and reproducibility. Second, more recent approaches, such as generative models (e.g., TabDDPM) or algorithm-level adjustments (e.g., Focal Loss), introduce significant complexity and computational overhead. Given the extensive scope of our study—encompassing three datasets and seven learning algorithms—the selected methods allow for a more controlled and comparable experimental setup. Finally, these four techniques collectively represent distinct philosophies in data balancing (random removal, density-agnostic augmentation, boundary-focused augmentation, and learning-difficulty-based augmentation), allowing for a broad-spectrum analysis of their effects on model performance.

3.3. Classification of Balanced Data

In today's world, data is being generated at an unprecedented scale. There is a constant flow of data from everywhere—smartphones, social media interactions, industrial sensors, and scientific experiments. Making sense of this massive data pile, gaining insights, and making smarter decisions require new tools. In this context, machine learning and its subfield, deep learning, come into play. These technologies enable computers to learn from data and perform tasks without human intervention, transforming many sectors.

Machine learning is based on the ability of computer systems to automatically perform specific tasks by learning from data [10]. In traditional programming, people set the rules and the system operates according to those rules. However, in machine learning, the system analyzes the data provided to it, recognizes patterns, and adapts to new data based on this knowledge. For example, an email spam filter can effectively classify incoming messages by learning from spam and normal messages.

Deep learning, on the other hand, is a specialized and often more powerful branch of machine learning. This approach is based on the artificial neural network (ANN) architecture, which is inspired by the neural connections in the human brain [14]. The word "deep" refers to the fact that these networks typically have many layers. Each layer processes the output of the previous one, allowing the network to learn increasingly complex and abstract representations of the data. While traditional ML methods require feature engineering, deep learning models can automatically extract features from large datasets.

The strength of deep learning is particularly evident in unstructured data types such as images, audio, and natural language. It has achieved significant success in areas like

image recognition (e.g., recognizing objects or faces in photos), natural language processing (text translation, sentiment analysis, chatbots), speech recognition (virtual assistants), and autonomous vehicles [45]. Deep learning shows high performance in complex detection and pattern recognition tasks where traditional ML algorithms struggle. Therefore, especially in cases where large datasets and high computing power are available, deep learning has become an indispensable tool for solving complex problems.

In this study, in order to compare classification performances, five machine learning algorithms AdaBoost, Logistic Regression, Random Forest, XGBoost, and K-Nearest Neighbor and two deep learning algorithms Convolutional Neural Networks and Deep Neural Networks were applied. These models are listed below with their descriptions.

The selected model set was designed to cover diverse learning biases while keeping the experimental space manageable and directly comparable across three datasets, multiple task settings, and several balancing configurations. Logistic Regression was included as a linear and interpretable baseline, while KNN was used to represent distance-based local decision behavior. Random Forest, AdaBoost, and XGBoost were selected as complementary ensemble-based classical machine learning methods that are widely used for structured tabular intrusion-detection data. CNN and DNN were included to examine whether higher-capacity deep learning models offer an advantage under the same balancing conditions. More recent models such as LightGBM or Transformer-based architectures were not included because the aim of this study was not to conduct an exhaustive state-of-the-art competition, but to evaluate how foundational balancing strategies interact with representative and widely adopted model families under a controlled benchmark setting.

AdaBoost (AB): AdaBoost is an ensemble learning method that combines multiple weak learners to improve classification performance by iteratively focusing on previously misclassified samples [46]. In this study, it was included as a representative boosting-based classical machine learning model for tabular intrusion-detection data. Its inclusion also allows us to observe how error-focused boosting behavior responds to different balancing strategies.

K-Nearest Neighbor (KNN): KNN is an instance-based learning algorithm that classifies a sample according to the labels of its nearest neighbors in the feature space [47]. It was included to represent distance-based local decision behavior. Since KNN is sensitive to local sample distributions, it is also useful for examining how resampling changes neighborhood structure.

Logistic Regression (LR): Logistic Regression is a linear classification model that estimates class probabilities through a logistic decision function [48]. It was used as an interpretable baseline to represent linear decision behavior under different balancing settings. Its simplicity makes it useful for assessing whether the benefit of balancing is visible even in low-complexity classifiers.

Random Forest (RF): Random Forest is a tree-based ensemble method that combines multiple decision trees trained on randomized feature subsets and bootstrap samples [49]. It was selected because it is widely used and typically performs strongly on structured tabular security data. In addition, it provides a robust benchmark for evaluating whether balancing improves minority-class detection without sacrificing overall stability.

XGBoost (XG): XGBoost is a gradient boosting framework that builds decision trees sequentially to correct the errors of earlier trees [50]. It was selected as a strong ensemble-based benchmark for tabular intrusion-detection tasks. Its inclusion helps assess whether balancing remains beneficial even when a high-capacity boosting model is used.

Convolutional Neural Network (CNN): CNN is a deep learning architecture that learns hierarchical feature representations through convolution and pooling operations [51]. In this study, it was included as a higher-capacity deep learning baseline to examine whether

deep feature extraction improves detection performance under different balancing settings. It also enables comparison between classical machine learning models and feature-learning-based models under the same evaluation protocol.

Deep Neural Network (DNN): DNN is a multilayer feedforward neural network that learns increasingly abstract feature representations through stacked hidden layers [31]. It was included as a general deep learning baseline for comparison with classical machine learning models under the same experimental protocol. This model is particularly useful for examining whether balancing affects deep fully connected architectures differently from tree-based or distance-based methods.

3.4. Experimental Setup and Reproducibility

All experiments were conducted using a fixed random seed protocol and repeated over multiple seeds to reduce the effect of random variation. For each dataset, the data were divided into training and test partitions using stratified splitting. The maximum number of epochs, batch sizes, optimizer settings, and early stopping criteria are summarized in Table 6.

To improve reproducibility, we report the main hyperparameter settings for all models, including the balancing strategies, KNN, Logistic Regression, Random Forest, XGBoost, CNN, and DNN configurations. We also document the evaluation scenarios, random seeds, and feature set used in the experiments. Since the purpose of the study is comparative benchmarking rather than extensive hyperparameter optimization, we adopted a consistent configuration strategy across datasets and balancing settings.

To assess result stability, all experiments were repeated over multiple random seeds, and the reported summaries include mean, standard deviation, and confidence-oriented descriptive statistics. In addition, pairwise non-parametric significance testing was performed to compare each balancing scenario against the original-train/original-test baseline for the same model, providing statistical support for the main performance comparisons.

Table 6. Main hyperparameter settings and experimental configuration.

Component	Setting	Value
Experimental protocol	Random seeds	42, 52, 62
	Train/test split	Stratified split, 80%/20%
	Evaluation scenarios	Original Tr./Original Te.; Original Tr./Balanced Te.; Balanced Tr. (RUS-SMOTE)/Original Te.; Balanced Tr. (RUS-SMOTE)/Balanced Te.; Balanced Tr. (RUS-B.SMOTE)/Original Te.; Balanced Tr. (RUS-B.SMOTE)/Balanced Te.; Balanced Tr. (RUS-ADASYN)/Original Te.; Balanced Tr. (RUS-ADASYN)/Balanced Te.
Balancing	Original	No balancing
	RUS-SMOTE	RandomUnderSampler + SMOTE
	RUS-B.SMOTE	RandomUnderSampler + Borderline-SMOTE
	RUS-ADASYN	RandomUnderSampler + ADASYN
AdaBoost	Base estimator	DecisionTreeClassifier()
	Number of estimators	50
	Learning rate	1.0
	Algorithm	SAMME
KNN	Number of neighbors	3
	Leaf size	9
	Distance metric	Euclidean

Table 6. *Cont.*

Component	Setting	Value
Logistic Regression	Penalty	L2
	Solver	lbfgs
	C	1.0
	Maximum iterations	1000
Random Forest	Number of trees	100
	Criterion	Gini
	Max features	sqrt
XGBoost	Objective	binary:logistic/multi:softprob
	Evaluation metric	logloss/mlogloss
	Number of estimators	100
	Learning rate	0.1
	Maximum depth	6
CNN	Architecture	Conv1D(32,3) → MaxPooling1D(2) → Conv1D(64,3) → MaxPooling1D(2) → Flatten → Dense(128) → Dense(softmax)
	Optimizer	Adam
	Batch size	32
	Maximum epochs	50
	Early stopping patience	5
DNN	Architecture	Dense(32) → Dropout(0.025) → Dense(32) → Dropout(0.025) → Dense(32) → Dropout(0.025) → Dense(softmax)
	Optimizer	Adam (learning rate = 0.001)
	L2 regularization	0.0001
	Batch size	42
	Maximum epochs	50
	Early stopping patience	5

4. Results and Discussion

In this section, we examine the effects of balancing procedures on the performance of machine learning and deep learning models, conducted on the WUSTL, ECU-IoHT, and CICIoMT2024 datasets used in our study. The procedures and evaluations were performed separately for both binary and multiclass structures present in the datasets. The interpretation of the evaluation results is based on both overall and class-sensitive metrics, including Accuracy, Precision, Recall, F1-Score, PR-AUC, FNR, minority-class recall, and cost-oriented missed-detection summaries. In this way, the discussion reflects not only general predictive performance but also the operational security implications of missed minority-class attacks. The results are presented in tables, with the bolded values indicating the highest performance achieved in each test scenario.

4.1. Metrics

To evaluate the models under both overall and class-sensitive perspectives, we report Accuracy, Precision, Recall, and F1-Score together with additional diagnostic metrics, namely PR-AUC, macro false positive rate (FPR), macro false negative rate (FNR), and minority-class recall. These metrics are particularly relevant for IoMT intrusion detection because strong overall performance may still hide missed detections in minority attack classes. In particular, PR-AUC is informative under class imbalance, while FNR and minority-class recall directly reflect the risk of under-detecting rare attacks. Accordingly,

the evaluation protocol combines global performance indicators with class-sensitive metrics to provide a more security-relevant interpretation of the results.

Accuracy is the ratio of correctly classified samples to the total number of samples. As shown in (1), the sum of true positives and true negatives is divided by the total number of cases. This metric shows the overall correctness of the model. However, if there is an imbalance between classes (e.g., one class is much larger than the other), this metric can be misleading.

$$\text{Accuracy} = \frac{\text{TP} + \text{TN}}{\text{TP} + \text{TN} + \text{FP} + \text{FN}} \tag{1}$$

Precision, as shown in (2), is the ratio of truly positive samples among those predicted as positive. It is important in cases where the cost of false positives is high (e.g., spam detection).

$$\text{Precision} = \frac{\text{TP}}{\text{TP} + \text{FP}} \tag{2}$$

Recall is the ratio of correctly predicted positive samples among all actual positive samples, as shown in (3).

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

F1-Score is the harmonic mean of precision and recall, as in (4), and provides a balance between these two metrics. The F1-Score is preferred when both false positives and false negatives are important, or when classes are imbalanced.

$$\text{F1-Score} = 2 \times \frac{\text{Precision} \times \text{Recall}}{\text{Precision} + \text{Recall}} \tag{4}$$

In Equations (1)–(4), TP, TN, FP, and FN denote true positives, true negatives, false positives, and false negatives, respectively.

4.2. Impact of Balancing Strategies on Binary Classification Performance

In this section, the binary classification results obtained on the WUSTL, ECU-IoHT, and CICIoMT2024 datasets are presented. In this approach, the various attack types present in the datasets were combined under a single "attack" category, while normal traffic was considered as a separate class.

When the results of the experimental studies conducted on the WUSTL dataset are examined through Tables 7 and 8, significant differences are observed in the performance of the algorithms and the effects of data balancing techniques. The XGBoost algorithm achieved the highest accuracy and F1-Score values with the original training and test data, exhibiting higher performance compared to other algorithms. This result shows that XGBoost's gradient boosting technique is effective in capturing complex patterns in the WUSTL dataset.

Table 7. WUSTL Accuracy—2 Classes: Accuracy Results Obtained from the Proposed Method.

Test	AB	CNN	DNN	KNN	LR	RF	XG
Original Tr. Original Te.	**0.9449**	**0.9298**	**0.9308**	**0.9317**	**0.9449**	**0.9360**	**0.9562**
Original Tr. Balanced Te.	0.8762	0.7260	0.7464	0.8293	0.8762	0.7500	0.8582
Balanced Tr. (RUS-SMOTE) Original Te.	0.9231	0.7534	0.8811	0.8618	0.9231	0.9093	0.9433
Balanced Tr. (RUS-SMOTE) Balanced Te.	0.8858	0.8149	0.7921	0.8606	0.8858	0.8353	0.9135
Balanced Tr. (RUS-B.SMOTE) Original Te.	0.9170	0.8615	0.6982	0.8514	0.9170	0.9059	0.9338
Balanced Tr. (RUS-B.SMOTE) Balanced Te.	0.8762	0.8065	0.7837	0.8606	0.8762	0.8498	0.9111
Balanced Tr. (RUS-ADASYN) Original Te.	0.9099	0.7914	0.7169	0.8416	0.9099	0.8986	0.9259
Balanced Tr. (RUS-ADASYN) Balanced Te.	0.8894	0.8137	0.7981	0.8630	0.8894	0.8558	0.9038

Bold values indicate the highest result in each test scenario. Tr: Train, Te: Test, RUS: RandomUnderSampler, B.SMOTE: Borderline-SMOTE.

Table 8. WUSTL F1-Score—2 Classes: F1-Score Results Obtained from the Proposed Method.

Test Name	AB	CNN	DNN	KNN	LR	RF	XG
Original Tr. Original Te.	**0.9446**	**0.9180**	**0.9199**	**0.9305**	**0.9446**	**0.9270**	**0.9541**
Original Tr. Balanced Te.	0.8751	0.7058	0.7293	0.8261	0.8751	0.7333	0.8558
Balanced Tr. (RUS-SMOTE) Original Te.	0.9272	0.7926	0.8883	0.8771	0.9272	0.9120	0.9450
Balanced Tr. (RUS-SMOTE) Balanced Te.	0.8856	0.8146	0.7910	0.8606	0.8856	0.8337	0.9132
Balanced Tr. (RUS-B.SMOTE) Original Te.	0.9219	0.8755	0.7483	0.8691	0.9219	0.9114	0.9374
Balanced Tr. (RUS-B.SMOTE) Balanced Te.	0.8761	0.8050	0.7820	0.8606	0.8761	0.8492	0.9110
Balanced Tr. (RUS-ADASYN) Original Te.	0.9163	0.8220	0.7635	0.8619	0.9163	0.9060	0.9303
Balanced Tr. (RUS-ADASYN) Balanced Te.	0.8892	0.8136	0.7973	0.8629	0.8892	0.8554	0.9038

Bold values indicate the highest result in each test scenario. Tr: Train, Te: Test, RUS: RandomUnderSampler, B.SMOTE: Borderline-SMOTE.

When the impact of data balancing techniques is analyzed, a clear decrease in performance was observed across all algorithms when using balanced test data with the original training data. This drop was particularly high for the CNN algorithm, with its accuracy decreasing from 0.92 to 0.72. This indicates that when CNN is trained on an imbalanced dataset, its ability to generalize on balanced test data is limited. Similarly, the DNN algorithm was also adversely affected by the balancing procedures, with its performance dropping from 0.93 to 0.74. When RUS-SMOTE balanced training data and original test data were used, XGBoost and AdaBoost/LR still showed high performance. This result demonstrates that these algorithms are able to adapt to the original data distribution even when trained on balanced data. On the other hand, the CNN algorithm also exhibited low performance in this scenario, suggesting that deep learning models are more sensitive to data balancing techniques.

When comparing the balancing techniques, RUS-SMOTE generally yielded better results than the others. The RUS-B.SMOTE and RUS-ADASYN techniques resulted in lower performance, especially for deep learning models. The DNN algorithm, when trained with RUS-B.SMOTE balanced data and tested on original data, achieved only 0.69 accuracy. This shows that the Borderline-SMOTE technique, which is based on generating synthetic border samples, was unable to properly represent class boundaries in the WUSTL dataset. The KNN algorithm showed relatively stable performance across different balancing scenarios. This suggests that the instance-based nature of KNN makes it less affected by data balancing techniques. However, the performance of KNN still lagged behind algorithms like XGBoost and AdaBoost.

Overall, it was observed that training with the original data yielded better results than training with balanced data. This indicates that the imbalance in the WUSTL dataset does not significantly affect algorithm performance and that balancing procedures can even lead to performance degradation.

When the results of the experimental studies conducted on the ECU-IoHT dataset are examined through Tables 9 and 10, it is observed that all algorithms and all balancing scenarios exhibit high performance. For the ECU-IoHT dataset, very high accuracy and F1-Score values were achieved for all algorithms and all balancing scenarios. This indicates that the features representing attack and normal classes in the dataset are very distinct and easily separable.

When the impact of balancing procedures on performance is analyzed, it is seen that the effect of balancing on the ECU-IoHT dataset is minimal. This is mainly due to the already high initial performance values. Nevertheless, some balancing scenarios resulted in very slight improvements. With the RUS-ADASYN balancing technique, the RF and XG algorithms achieved perfect accuracy on the balanced test data. Similarly, with RUS-ADASYN balanced training and original test data, the AB and LR algorithms also achieved perfect accuracy. When comparing algorithm performances, it is seen that all algorithms

perform similarly on the ECU-IoHT dataset. This indicates that the dataset is highly suitable for classification, and algorithm selection is not a critical factor.

Table 9. ECU-IoHT Accuracy—2 Classes: Accuracy Results Obtained from the Proposed Method.

Test Name	AB	CNN	DNN	KNN	LR	RF	XG
Original Tr. Original Te.	0.9999	0.9988	0.9986	0.9996	0.9999	0.9999	0.9999
Original Tr. Balanced Te.	0.9998	0.9952	0.9976	0.9997	0.9998	0.9999	0.9999
Balanced Tr. (RUS-SMOTE) Original Te.	0.9999	0.9984	0.9985	0.9996	0.9999	0.9999	0.9999
Balanced Tr. (RUS-SMOTE) Balanced Te.	0.9999	0.9993	0.9991	0.9998	0.9999	0.9999	0.9999
Balanced Tr. (RUS-B.SMOTE) Original Te.	0.9998	0.9991	0.9984	0.9995	0.9998	0.9999	0.9999
Balanced Tr. (RUS-B.SMOTE) Balanced Te.	0.9996	0.9994	0.9990	0.9994	0.9996	0.9999	0.9999
Balanced Tr. (RUS-ADASYN) Original Te.	**1.0000**	0.9986	0.9986	0.9998	**1.0000**	0.9999	0.9999
Balanced Tr. (RUS-ADASYN) Balanced Te.	0.9999	**0.9996**	**0.9993**	**0.9999**	0.9999	**1.0000**	**1.0000**

Bold values indicate the highest result in each test scenario. Tr: Train, Te: Test, RUS: RandomUnderSampler, B.SMOTE: Borderline-SMOTE.

Table 10. ECU-IoHT F1-Score—2 Classes: F1-Score Results Obtained from the Proposed Method.

Test Name	AB	CNN	DNN	KNN	LR	RF	XG
Original Tr. Original Te.	0.9999	0.9988	0.9986	0.9996	0.9999	0.9999	0.9999
Original Tr. Balanced Te.	0.9998	0.9952	0.9976	0.9997	0.9998	0.9999	0.9999
Balanced Tr. (RUS-SMOTE) Original Te.	0.9999	0.9984	0.9985	0.9996	0.9999	0.9999	0.9999
Balanced Tr. (RUS-SMOTE) Balanced Te.	0.9999	0.9993	0.9991	0.9998	0.9999	0.9999	0.9999
Balanced Tr. (RUS-B.SMOTE) Original Te.	0.9998	0.9991	0.9984	0.9995	0.9998	0.9999	0.9999
Balanced Tr. (RUS-B.SMOTE) Balanced Te.	0.9996	0.9994	0.9990	0.9994	0.9996	0.9999	0.9999
Balanced Tr. (RUS-ADASYN) Original Te.	**1.0000**	0.9986	0.9986	0.9998	**1.0000**	0.9999	0.9999
Balanced Tr. (RUS-ADASYN) Balanced Te.	0.9999	**0.9996**	**0.9993**	**0.9999**	0.9999	**1.0000**	**1.0000**

Bold values indicate the highest result in each test scenario. Tr: Train, Te: Test, RUS: RandomUnderSampler, B.SMOTE: Borderline-SMOTE.

Even more complex models like CNN and DNN achieved the same performance as simpler algorithms, showing that the problem does not require complex models. When comparing the balancing techniques, RUS-ADASYN was observed to give slightly better results than the other techniques, but since all balancing methods performed similarly, the choice of balancing technique is not critical for this dataset. Overall, it can be concluded that the structure of the ECU-IoHT dataset is highly suitable for classification, and the choice of algorithm or balancing technique does not have a significant impact on performance.

When the results of the experimental studies conducted on the CICIoMT2024 dataset are examined through Tables 11 and 12, it is observed that the effect of balancing procedures on performance is most pronounced in this dataset. In general, high accuracy and F1-Score values were achieved with the CICIoMT2024 dataset. However, compared to the ECU-IoHT dataset, slightly lower and more variable results were observed. This shows that the classes in the CICIoMT2024 dataset are not as clearly separated as those in the ECU-IoHT dataset.

Table 11. CICIoMT2024 Accuracy—2 Classes: Accuracy Results Obtained from the Proposed Method.

Test Name	AB	CNN	DNN	KNN	LR	RF	XG
Original Tr. Balanced Te.	0.9727	0.9503	0.9444	0.9475	0.9727	0.9792	0.9756
Balanced Tr. (RUS-SMOTE) Original Te.	0.9985	0.9966	0.9960	0.9964	0.9985	**0.9987**	0.9986
Balanced Tr. (RUS-SMOTE) Balanced Te.	0.9861	0.9840	0.9866	0.9721	0.9861	0.9928	0.9895
Balanced Tr. (RUS-B.SMOTE) Original Te.	**0.9989**	0.9961	**0.9963**	0.9962	**0.9989**	0.9986	**0.9989**
Balanced Tr. (RUS-B.SMOTE) Balanced Te.	0.9921	0.9810	0.9948	0.9667	0.9921	0.9908	0.9986
Balanced Tr. (RUS-ADASYN) Original Te.	0.9985	**0.9968**	0.9959	**0.9965**	0.9985	**0.9987**	0.9988
Balanced Tr. (RUS-ADASYN) Balanced Te.	0.9848	0.9944	0.9883	0.9755	0.9848	0.9926	0.9968

Bold values indicate the highest result in each test scenario. Tr: Train, Te: Test, RUS: RandomUnderSampler, B.SMOTE: Borderline-SMOTE.

Table 12. CICIoMT2024 F1-Score—2 Classes: F1-Score Results Obtained from the Proposed Method.

Test Name	AB	CNN	DNN	KNN	LR	RF	XG
Original Tr. Balanced Te.	0.9727	0.9502	0.9442	0.9473	0.9727	0.9792	0.9756
Balanced Tr. (RUS-SMOTE) Original Te.	0.9985	0.9967	0.9961	0.9964	0.9985	**0.9987**	0.9986
Balanced Tr. (RUS-SMOTE) Balanced Te.	0.9861	0.9840	0.9866	0.9721	0.9861	0.9928	0.9895
Balanced Tr. (RUS-B.SMOTE) Original Te.	**0.9989**	0.9962	**0.9964**	0.9963	**0.9989**	**0.9987**	**0.9989**
Balanced Tr. (RUS-B.SMOTE) Balanced Te.	0.9921	0.9810	0.9948	0.9667	0.9921	0.9908	0.9986
Balanced Tr. (RUS-ADASYN) Original Te.	0.9985	**0.9969**	0.9960	**0.9966**	0.9985	**0.9987**	0.9988
Balanced Tr. (RUS-ADASYN) Balanced Te.	0.9848	0.9944	0.9883	0.9755	0.9848	0.9926	0.9968

Bold values indicate the highest result in each test scenario. Tr: Train, Te: Test, RUS: RandomUnderSampler, B.SMOTE: Borderline-SMOTE.

When original training and balanced test data were used, performance ranged between 0.94 and 0.97, whereas using balanced training and original test data, all algorithms achieved high values such as 0.99. This indicates that balancing, especially when applied to the training data, improves the generalization ability of the models.

When comparing the balancing techniques, it was observed that all techniques (RUS-SMOTE, RUS-B.SMOTE, RUS-ADASYN) showed similar performance. RUS-B.SMOTE gave slightly better results than the other techniques with balanced training and balanced test data. This shows that the choice of balancing technique is not a critical factor for the CICIoMT2024 dataset.

Overall, it can be concluded that balancing procedures, especially when applied to the training data, significantly improve the performance of algorithms for the CICIoMT2024 dataset. This demonstrates that the imbalance in the CICIoMT2024 dataset affects algorithm performance and that balancing helps mitigate this effect.

The binary classification results show that the effectiveness of data balancing techniques depends significantly on the specific characteristics of the dataset and the structure of the model. In the WUSTL dataset, it was observed that the original imbalanced structure provided better generalization for some models, while in the ECU-IoHT dataset, balancing procedures did not significantly affect performance. In the CICIoMT2024 dataset, certain balancing methods improved the performance of some models by emphasizing minority classes. However, when both the training and test sets were balanced, a decrease in performance was generally observed, highlighting the need for careful application of balancing strategies and consideration of their potential adverse effects on test data.

4.3. Impact of Balancing Strategies on Multiclass Classification Performance

Among the datasets used in the study, ECU-IoHT and CICIoMT2024 are suitable for multiclass classification tasks. The ECU-IoHT dataset contains five categories, while the CICIoMT2024 dataset includes six different categories. Additionally, only the CICIoMT2024 dataset covers data for a total of eighteen different attack types.

For the ECU-IoHT dataset, the models in the five-class structure achieved high accuracy values (see Tables 13 and 14). Similar to the binary structure, in the multiclass setup, balancing procedures did not significantly change model performance. In balanced training and test sets created with RandomUnderSampler-SMOTE, RandomUnderSampler-Borderline-SMOTE, and RandomUnderSampler-ADASYN, performance generally remained stable or showed only very slight decreases. All models achieved F1-Scores close to 1.00 even on the original dataset. This result confirms that the ECU-IoHT dataset has highly distinctive features and that balancing does not provide additional benefit.

For the CICIoMT2024 dataset, in both the 6-class and 19-class structures, the effect of balancing procedures was again significant, but a consistent performance improvement could not be achieved (see Tables 15–18). In the six-class structure of the CICIoMT2024 dataset, accuracy and F1-Score values ranged between 0.57 and 0.99 overall. These values

are lower and more variable compared to the results obtained in the binary structure. This indicates that as the number of classes increases, the classification problem becomes more difficult. Increasing the number of classes makes the distinction between classes more complex and causes the performance of the algorithms to decrease.

Table 13. ECU-IoHT Accuracy—5 Classes: Accuracy Results Obtained from the Proposed Method.

Test Name	AB	CNN	DNN	KNN	LR	RF	XG
Original Tr. Original Te.	0.9999	0.9989	0.9984	0.9996	0.9999	0.9999	**1.0000**
Original Tr. Balanced Te.	**1.0000**	0.9985	0.9985	0.9985	**1.0000**	**1.0000**	**1.0000**
Balanced Tr. (RUS-SMOTE) Original Te.	**1.0000**	0.9983	0.9985	0.9996	**1.0000**	**1.0000**	0.9998
Balanced Tr. (RUS-SMOTE) Balanced Te.	**1.0000**	**1.0000**	**1.0000**	0.9985	**1.0000**	**1.0000**	**1.0000**
Balanced Tr. (RUS-B.SMOTE) Original Te.	**1.0000**	0.9985	0.9983	0.9996	**1.0000**	**1.0000**	0.9999
Balanced Tr. (RUS-B.SMOTE) Balanced Te.	**1.0000**	0.9985	0.9985	0.9985	**1.0000**	**1.0000**	0.9985
Balanced Tr. (RUS-ADASYN) Original Te.	**1.0000**	0.9983	0.9984	0.9996	**1.0000**	0.9999	0.9999
Balanced Tr. (RUS-ADASYN) Balanced Te.	**1.0000**	**1.0000**	0.9985	**1.0000**	**1.0000**	**1.0000**	**1.000**

Bold values indicate the highest result in each test scenario. Tr: Train, Te: Test, RUS: RandomUnderSampler, B.SMOTE: Borderline-SMOTE.

Table 14. ECU-IoHT F1-Score—5 Classes: F1-Score Results Obtained from the Proposed Method.

Test Name	AB	CNN	DNN	KNN	LR	RF	XG
Original Tr. Original Te.	0.9999	0.9989	0.9984	0.9996	0.9999	0.9999	**1.0000**
Original Tr. Balanced Te.	**1.0000**	0.9985	0.9985	0.9985	**1.0000**	**1.0000**	**1.0000**
Balanced Tr. (RUS-SMOTE) Original Te.	**1.0000**	0.9983	0.9985	0.9996	**1.0000**	**1.0000**	0.9998
Balanced Tr. (RUS-SMOTE) Balanced Te.	**1.0000**	**1.0000**	**1.0000**	0.9985	**1.0000**	**1.0000**	**1.0000**
Balanced Tr. (RUS-B.SMOTE) Original Te.	**1.0000**	0.9985	0.9983	0.9996	**1.0000**	**1.0000**	0.9999
Balanced Tr. (RUS-B.SMOTE) Balanced Te.	**1.0000**	0.9985	0.9985	0.9985	**1.0000**	**1.0000**	0.9985
Balanced Tr. (RUS-ADASYN) Original Te.	**1.0000**	0.9983	0.9984	0.9996	**1.0000**	0.9999	0.9999
Balanced Tr. (RUS-ADASYN) Balanced Te.	**1.0000**	**1.0000**	0.9985	**1.0000**	**1.0000**	**1.0000**	**1.0000**

Bold values indicate the highest result in each test scenario. Tr: Train, Te: Test, RUS: RandomUnderSampler, B.SMOTE: Borderline-SMOTE.

Table 15. CICIoMT2024 Accuracy—6 Classes: Accuracy Results Obtained from the Proposed Method.

Test Name	AB	CNN	DNN	KNN	LR	RF	XG
Original Tr. Balanced Te.	0.9532	**0.8620**	0.7480	0.8801	0.9532	0.9620	0.9601
Balanced Tr. (RUS-SMOTE) Original Te.	0.9971	0.6695	0.6360	0.9106	0.9971	0.9980	0.9970
Balanced Tr. (RUS-SMOTE) Balanced Te.	0.9548	0.8172	**0.8050**	0.8734	0.9548	0.9644	0.9542
Balanced Tr. (RUS-B.SMOTE) Original Te.	0.9975	0.6411	0.6239	**0.9115**	0.9975	**0.9985**	0.9967
Balanced Tr. (RUS-B.SMOTE) Balanced Te.	0.9580	0.7744	0.7383	0.8571	0.9580	0.9595	0.9288
Balanced Tr. (RUS-ADASYN) Original Te.	**0.9979**	0.5798	0.6284	0.9094	**0.9979**	0.9983	**0.9975**
Balanced Tr. (RUS-ADASYN) Balanced Te.	0.9565	0.7784	0.7440	0.8573	0.9565	0.9604	0.9357

Bold values indicate the highest result in each test scenario. Tr: Train, Te: Test, RUS: RandomUnderSampler, B.SMOTE: Borderline-SMOTE.

Table 16. CICIoMT2024 Accuracy—19 Classes: Accuracy Results Obtained from the Proposed Method.

Test Name	AB	CNN	DNN	KNN	LR	RF	XG
Original Tr. Balanced Te.	0.8667	**0.7071**	0.5580	0.7852	0.8667	0.9041	0.8633
Balanced Tr. (RUS-SMOTE) Original Te.	0.9950	0.5526	0.5989	0.9029	0.9950	**0.9920**	0.9922
Balanced Tr. (RUS-SMOTE) Balanced Te.	0.9295	0.6780	**0.6709**	0.7988	0.9295	0.8766	0.9052
Balanced Tr. (RUS-B.SMOTE) Original Te.	**0.9953**	0.6855	0.6415	**0.9197**	**0.9953**	0.9919	**0.9954**
Balanced Tr. (RUS-B.SMOTE) Balanced Te.	0.9174	0.6607	0.6500	0.7934	0.9174	0.8741	0.9363
Balanced Tr. (RUS-ADASYN) Original Te.	0.9930	0.5664	0.5520	0.9109	0.9930	0.9918	**0.9954**
Balanced Tr. (RUS-ADASYN) Balanced Te.	0.9046	0.6590	0.6568	0.7980	0.9046	0.8727	0.9211

Bold values indicate the highest result in each test scenario. Tr: Train, Te: Test, RUS: RandomUnderSampler, B.SMOTE: Borderline-SMOTE.

When analyzing the effect of balancing, it was observed that balancing procedures had a significant impact in the six-class structure of the CICIoMT2024 dataset, but a consistent performance improvement could not be achieved. The combination of balanced training and original test data generally yielded the highest performance. In particular, AdaBoost, Logistic Regression, Random Forest, and XGBoost achieved high accuracy and F1-Score values, such as 0.99, with this combination. This shows that balancing procedures, especially when applied to the training data, improve the generalization ability of the models. However, with the combination of balanced training and balanced test data, algorithm performance decreased. For example, AdaBoost and Logistic Regression achieved 0.99 accuracy with balanced training and original test data, but only 0.95 accuracy with balanced training and balanced test data. This suggests that applying balancing procedures to the test data can disrupt the original data distribution and lead to decreased performance.

Table 17. CICIoMT2024 F1-Score—6 Classes: F1-Score Results Obtained from the Proposed Method.

Test Name	AB	CNN	DNN	KNN	LR	RF	XG
Original Tr. Balanced Te.	0.9524	**0.8465**	0.7279	0.8756	0.9524	0.9615	0.9600
Balanced Tr. (RUS-SMOTE) Original Te.	0.9972	0.6845	0.6546	0.9129	0.9972	0.9980	0.9972
Balanced Tr. (RUS-SMOTE) Balanced Te.	0.9543	0.8169	**0.8056**	0.8707	0.9543	0.9641	0.9536
Balanced Tr. (RUS-B.SMOTE) Original Te.	0.9976	0.6589	0.6428	**0.9137**	0.9976	**0.9985**	0.9968
Balanced Tr. (RUS-B.SMOTE) Balanced Te.	0.9574	0.7729	0.7350	0.8525	0.9574	0.9587	0.9271
Balanced Tr. (RUS-ADASYN) Original Te.	**0.9980**	0.5964	0.6465	0.9115	**0.9980**	0.9983	**0.9974**
Balanced Tr. (RUS-ADASYN) Balanced Te.	0.9560	0.7735	0.7411	0.8524	0.9560	0.9599	0.9337

Bold values indicate the highest result in each test scenario. Tr: Train, Te: Test, RUS: RandomUnderSampler, B.SMOTE: Borderline-SMOTE.

Table 18. CICIoMT2024 F1-Score—19 Classes: F1-Score Results Obtained from the Proposed Method.

Test Name	AB	CNN	DNN	KNN	LR	RF	XG
Original Tr. Balanced Te.	0.8515	0.6593	0.4676	0.7690	0.8515	0.9008	0.8541
Balanced Tr. (RUS-SMOTE) Original Te.	0.9951	0.5646	0.5904	0.9058	0.9951	**0.9910**	0.9923
Balanced Tr. (RUS-SMOTE) Balanced Te.	0.9289	0.6531	**0.6310**	0.7840	0.9289	0.8612	0.9003
Balanced Tr. (RUS-B.SMOTE) Original Te.	**0.9953**	**0.6729**	0.6184	**0.9214**	**0.9953**	0.9908	**0.9955**
Balanced Tr. (RUS-B.SMOTE) Balanced Te.	0.9167	0.6352	0.6177	0.7808	0.9167	0.8608	0.9347
Balanced Tr. (RUS-ADASYN) Original Te.	0.9929	0.5756	0.5538	0.9131	0.9929	0.9907	**0.9955**
Balanced Tr. (RUS-ADASYN) Balanced Te.	0.9031	0.6399	0.6297	0.7877	0.9031	0.8605	0.9187

Bold values indicate the highest result in each test scenario. Tr: Train, Te: Test, RUS: RandomUnderSampler, B.SMOTE: Borderline-SMOTE.

CNN and DNN showed significantly lower performance compared to other algorithms. In particular, CNN achieved low accuracy and F1-Score values (around 0.66–0.68) with balanced training and original test data. Similarly, DNN achieved low values (around 0.63–0.65). This indicates that deep learning models perform worse than traditional machine learning algorithms in multiclass structures, especially on balanced datasets. KNN showed moderate performance. KNN's performance was highest with balanced training and original test data, but lower with balanced training and balanced test data. The instance-based nature of KNN may cause a decrease in performance on balanced test data.

In the 19-class structure of the CICIoMT2024 dataset, accuracy and F1-Score values ranged between 0.46 and 0.99. These values are lower and more variable compared to those obtained in the six-class structure. This indicates that increasing the number of classes from six to nineteen makes the distinction between classes even more complex and decreases algorithm performance. Analyzing the effect of balancing, it was observed that balancing procedures were again important in the 19-class structure of the CICIoMT2024 dataset, but a consistent performance improvement could not be achieved.

The combination of balanced training and original test data generally yielded the highest performance. In particular, AdaBoost, Logistic Regression, Random Forest, and XGBoost achieved very high accuracy and F1-Score values (around 0.99) with this combination. This demonstrates that balancing, especially when applied to the training data, improves the generalization ability of models. However, with the combination of balanced training and balanced test data, algorithm performance decreased significantly. The combination of original training and balanced test data showed the lowest performance. In particular, the DNN algorithm achieved only 0.55 accuracy and 0.46 F1-Score with this combination.

CNN and DNN performed significantly worse than the strongest classical models in the 19-class setting. A plausible explanation is that the tabular network-flow structure of the dataset is more naturally aligned with tree-based ensemble learning than with the current CNN/DNN architectures. Moreover, synthetic resampling in highly heterogeneous multiclass settings may increase local class overlap and amplify borderline ambiguity, which can reduce the generalization ability of deep models more strongly than that of classical ensemble methods.

A plausible explanation for this behavior is that the tabular network-flow structure of the datasets is more naturally aligned with tree-based ensemble models than with the current CNN/DNN architectures. In addition, synthetic resampling in multi-class settings may increase local class overlap and amplify borderline ambiguity, which can reduce generalization for deep models more strongly than for classical ensemble methods. Therefore, the lower performance of CNN/DNN in some balanced multi-class scenarios should be interpreted as a dataset–model interaction effect rather than as evidence that deep learning is universally unsuitable for IoMT intrusion detection.

The methods applied to multiclass IoMT datasets have shown that the effects of data balancing techniques can be more complex and inconsistent compared to binary scenarios. In the five-class structure of the ECU-IoHT dataset, balancing procedures did not significantly alter the high accuracy levels of the models. However, in the six-class and nineteen-class structures of the CICIoMT2024 dataset, some balancing methods improved the performance of traditional machine learning models, while the use of balanced data in deep learning models (such as CNN and DNN) led to unexpected decreases in performance. This suggests that synthetic data generation methods may increase data complexity in multiclass and imbalanced datasets and may not be suitable for some types of models. Additionally, the generalization issues observed on balanced test sets emphasize that data balancing strategies should be carefully selected and evaluated in multiclass scenarios.

4.4. Class-Sensitive Diagnostic Analysis

The aggregate performance tables presented in the previous section provide a broad view of multiclass classification behavior. However, in IoMT intrusion detection, security-relevant failures are often hidden within minority attack classes that contribute little to overall metrics. This section therefore examines class-sensitive diagnostic indicators, including PR-AUC, minority-class recall, and false negative rate, as well as confusion-matrix-level error structures, to reveal detection weaknesses that aggregate scores alone cannot capture.

The diagnostic results in Table 19 show that aggregate F1-Score alone is insufficient to characterize multiclass IoMT intrusion detection performance. Recon-Ping_Sweep was selected as the hard-class diagnostic because it represents the smallest attack category in the CICIoMT2024 training set (740 samples) and therefore constitutes the most demanding minority-class detection scenario; its recall and FNR directly reflect the security cost of underrepresentation.

Under the original-test benchmark, the balanced-training configuration based on RUS-B.SMOTE clearly benefits the strongest tree-based models. XGBoost achieves the best combined profile across all three metrics, with an F1-Score of 0.9955, a macro PR-AUC of 0.938, and a Recon-Ping_Sweep recall of 0.8925, corresponding to an FNR of only 0.1075. Random Forest shows a similar improvement in aggregate F1-Score but lags noticeably behind XGBoost in hard-class recall (0.7258 vs. 0.8925), indicating that strong overall performance does not guarantee proportional minority-class recovery.

Table 19. Class-sensitive diagnostic results for the CICIoMT2024 19-class task. In the last two columns, the recall and FNR of the Recon-Ping_Sweep class are reported as a fixed hard-class diagnostic.

Model	Scenario	F1-Score	Macro PR-AUC	Recon-Ping_Sweep Recall	Recon-Ping_Sweep FNR
RF	Original Tr./Balanced Te.	0.9008	0.952	0.7688	0.2312
XG	Original Tr./Balanced Te.	0.8541	0.936	0.7581	0.2419
CNN	Original Tr./Balanced Te.	0.6593	0.816	0.4731	0.5269
DNN	Original Tr./Balanced Te.	0.4676	0.699	0.0000	1.0000
RF	Balanced Tr. (RUS-B.SMOTE)/Original Te.	0.9908	0.956	0.7258	0.2742
XG	Balanced Tr. (RUS-B.SMOTE)/Original Te.	**0.9955**	**0.938**	**0.8925**	**0.1075**
CNN	Balanced Tr. (RUS-B.SMOTE)/Original Te.	0.6729	0.655	0.8118	0.1882
DNN	Balanced Tr. (RUS-B.SMOTE)/Original Te.	0.6184	0.614	0.8710	0.1290

Bold values indicate the highest result in each test scenario. Tr: Train, Te: Test, RUS: RandomUnderSampler, B.SMOTE: Borderline-SMOTE.

A notable divergence is observed for the deep learning models under this scenario. Although CNN and DNN remain substantially weaker in aggregate F1-Score (0.6729 and 0.6184, respectively), their Recon-Ping_Sweep recall under balanced training exceeds that of Random Forest (0.8118 and 0.8710, respectively). This dissociation between aggregate and class-sensitive performance illustrates that deep learning models, despite their overall weakness in the multiclass setting, can recover specific minority classes more effectively than their global scores suggest. The low aggregate F1-Score of DNN therefore reflects heterogeneous performance across the full 19-class distribution rather than uniform failure across all attack types.

In contrast, the original-training/balanced-testing setting reveals the most critical failure: DNN achieves zero recall for Recon-Ping_Sweep (FNR = 1.000), meaning that this attack type is entirely missed when the model is trained on imbalanced data and evaluated under equal class pressure. This result confirms that training-set imbalance has the most severe consequence for minority-class detection, and that balancing the training data is a more security-critical design decision than the choice of test distribution. Overall, these findings confirm that multiclass balancing should be evaluated not only by aggregate performance but also by class-sensitive behavior on difficult minority attack types, and that the aggregate and class-sensitive metrics can tell meaningfully different stories for the same model.

As shown in Figure 2, the RF model preserves strong separation across most dominant TCP/IP attack categories, with near-perfect recall for several DDoS and DoS classes. The remaining errors are concentrated primarily in the reconnaissance-related categories and in a few minority classes such as malformed or spoofing traffic. In particular, the confusion among Recon-OS, Recon-Ping, Recon-Port, and Recon-VulScan indicates that these categories remain structurally more difficult to distinguish, even when balancing improves the overall multiclass profile.

In contrast, Figure 3 reveals substantially higher confusion for the CNN model, especially among reconnaissance and minority categories. Although some dominant classes remain well recognized, the recall values for several hard classes decrease markedly, and multiple minority attacks are redistributed across visually or statistically similar categories. This pattern is consistent with the lower F1-Score observed for CNN in Table 17 and sug-

gests that the multiclass degradation of deep learning models is concentrated in precisely those classes that are operationally important from a security perspective.

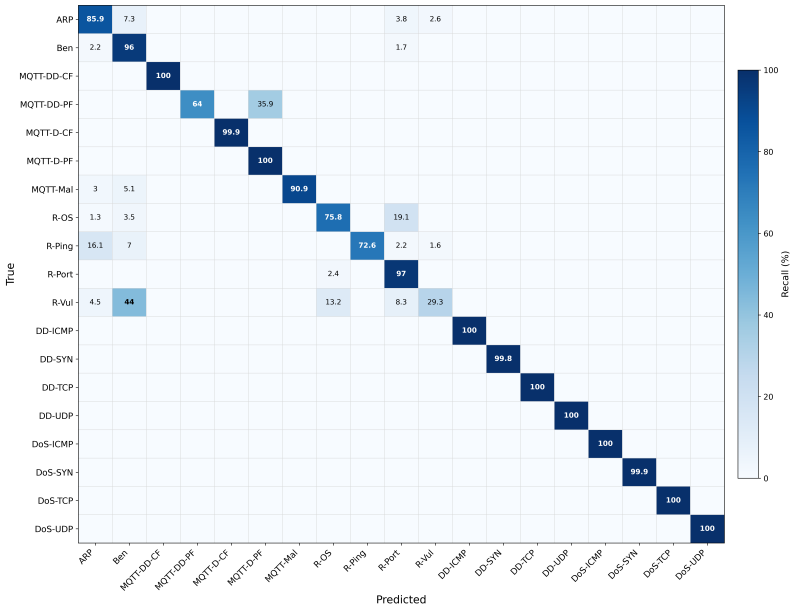


Figure 2. Row-normalized confusion matrix (%) for the RF model on the CICIoMT2024 19-class task under the Balanced Tr. (RUS-B.SMOTE)/Original Te. scenario. Abbreviations: DD = DDoS, R = Reconnaissance, Mal = Malformed, CF = Connect Flood, PF = Publish Flood.

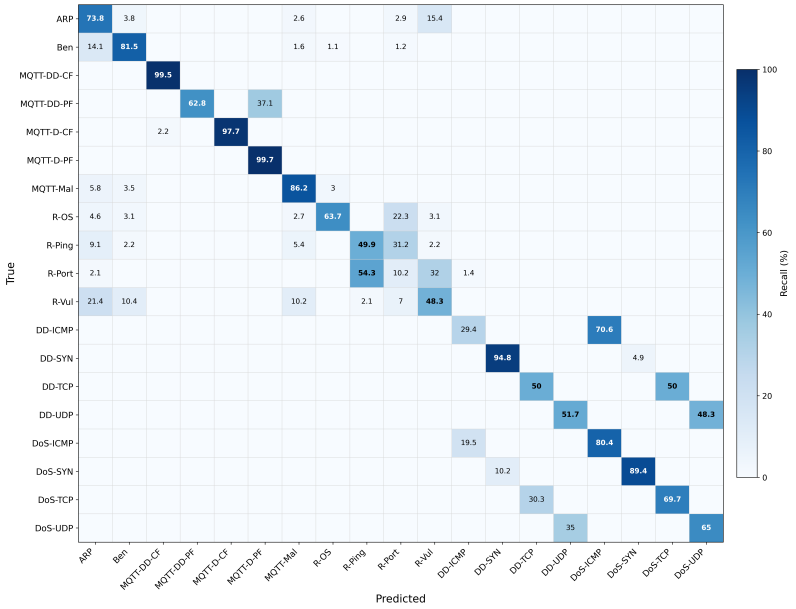


Figure 3. Row-normalized confusion matrix (%) for the CNN model on the CICIoMT2024 19-class task under the Balanced Tr. (RUS-B.SMOTE)/Original Te. scenario. Abbreviations: DD = DDoS, R = Reconnaissance, Mal = Malformed, CF = Connect Flood, PF = Publish Flood.

The comparison between Figures 2 and 3 shows that the performance gap between classical and deep learning models in this multiclass setting is not merely a small aggregate difference, but a qualitatively different error structure. While RF retains clearer decision boundaries across most attack categories, CNN exhibits broader inter-class confusion, particularly in the reconnaissance block and in several underrepresented classes. This visual evidence supports the conclusion that balancing alone is not sufficient to guarantee robust multiclass discrimination when the model architecture is not well aligned with the tabular flow-based structure of the data.

From a security perspective, these findings indicate that the benefit of balancing is dataset- and method-dependent rather than universal. In highly imbalanced settings such as CICIoMT2024, appropriate balancing strategies improved F1-Score and classification performance, suggesting a stronger ability to recognize underrepresented attack patterns. This is important for IoMT security because rare attack behaviors should not be ignored simply due to their low frequency in the training data. At the same time, some balancing configurations reduced performance, especially for certain deep learning models and balanced-test scenarios, showing that indiscriminate resampling may also weaken operational reliability. Therefore, balancing enhances IoMT security only when it improves robust detection of minority attack classes without introducing excessive class overlap or loss of generalization.

The per-attack analysis further shows that the security value of balancing should be interpreted through missed-detection behavior rather than overall performance alone. In particular, attack families with lower support, such as reconnaissance-oriented categories and selected MQTT flood variants, are more sensitive to the choice of balancing strategy. In these cases, reductions in false negatives and improvements in class-sensitive metrics are more meaningful from an operational IoMT security perspective than small changes in overall accuracy. Therefore, the practical benefit of balancing lies in improving the detectability of underrepresented but security-relevant attack behaviors rather than uniformly increasing all global metrics.

4.5. Statistical Significance of Balancing Strategies

To assess whether the observed F1-Score differences between balancing scenarios are statistically meaningful, we conducted Wilcoxon signed-rank tests on the WUSTL and ECU-IoHT datasets, for which results across three independent random seeds (42, 52, 62) were available. For each model, the F1-Score values obtained under each balanced scenario were compared against the Original Tr./Original Te. baseline using a two-sided Wilcoxon signed-rank test ($\alpha = 0.05$). Due to the computational cost of large-scale experiments on the CICIoMT2024 dataset (over 7 million training samples per run), three-seed repetition was not feasible within the revision period; accordingly, significance testing is reported only for WUSTL and ECU-IoHT.

Tables 20 and 21 report the F1-Score difference (ΔF1) and the associated p-value for each scenario–model combination. No statistically significant differences were detected in any comparison ($p > 0.05$ in all cases). For WUSTL, all p-values equal 0.250, which is the minimum achievable p-value for the Wilcoxon test with only three paired observations, indicating that the test is underpowered rather than that balancing has no effect. For ECU-IoHT, p-values are more varied due to the near-identical F1-Scores across seeds, reflecting the ceiling effect already observed in Table 10. These results confirm that the performance variations reported across scenarios fall within natural seed-to-seed variability and should not be interpreted as evidence of a statistically reliable advantage for any particular balancing strategy over the unbalanced baseline.

Table 20. Wilcoxon signed-rank test results for the WUSTL 2-class task.

Scenario	AB	RF	XG	CNN	DNN
Original Tr./Balanced Te.	−0.0615/0.250	−0.1785/0.250	−0.1620/0.250	−0.1583/0.250	−0.1693/0.250
Balanced Tr. (RUS-SMOTE)/Original Te.	−0.0248/0.250	−0.0099/0.250	−0.0195/0.250	−0.0700/0.250	−0.0548/0.250
Balanced Tr. (RUS-SMOTE)/Balanced Te.	−0.0611/0.250	−0.0806/0.250	−0.0406/0.250	−0.0932/0.250	−0.0990/0.250
Balanced Tr. (RUS-B.SMOTE)/Original Te.	−0.0383/0.250	−0.0176/0.250	−0.0524/0.250	−0.0927/0.250	−0.1065/0.250
Balanced Tr. (RUS-B.SMOTE)/Balanced Te.	−0.0693/0.250	−0.0665/0.250	−0.0485/0.250	−0.0869/0.250	−0.1018/0.250
Balanced Tr. (RUS-ADASYN)/Original Te.	−0.0409/0.250	−0.0234/0.250	−0.0565/0.250	−0.0722/0.250	−0.1156/0.250
Balanced Tr. (RUS-ADASYN)/Balanced Te.	−0.0621/0.250	−0.0611/0.250	−0.0498/0.250	−0.0901/0.250	−0.0909/0.250

Tr: Train, Te: Test, RUS: RandomUnderSampler, B.SMOTE: Borderline-SMOTE.

Table 21. Wilcoxon signed-rank test results for the ECU-IoHT 2-class task.

Scenario	AB	RF	XG	CNN	DNN
Original Tr./Balanced Te.	+0.0000/0.655	−0.0000/0.180	−0.0001/0.180	−0.0014/0.250	−0.0007/0.250
Balanced Tr. (RUS-SMOTE)/Original Te.	−0.0000/0.750	−0.0000/0.180	−0.0001/0.250	−0.0002/0.750	−0.0001/0.750
Balanced Tr. (RUS-SMOTE)/Balanced Te.	−0.0001/0.500	−0.0001/0.180	−0.0002/0.250	+0.0002/1.000	+0.0003/0.250
Balanced Tr. (RUS-B.SMOTE)/Original Te.	−0.0000/0.750	−0.0000/0.180	−0.0001/0.250	−0.0000/1.000	+0.0000/1.000
Balanced Tr. (RUS-B.SMOTE)/Balanced Te.	−0.0001/0.500	−0.0001/0.180	−0.0002/0.250	+0.0001/1.000	+0.0003/0.500
Balanced Tr. (RUS-ADASYN)/Original Te.	−0.0000/0.750	−0.0000/0.500	−0.0001/0.250	−0.0001/1.000	+0.0000/1.000
Balanced Tr. (RUS-ADASYN)/Balanced Te.	−0.0000/0.655	−0.0001/0.500	−0.0001/0.250	+0.0002/1.000	+0.0004/0.250

Tr: Train, Te: Test, RUS: RandomUnderSampler, B.SMOTE: Borderline-SMOTE.

4.6. Practical Guidance for Selecting Balancing Strategies

The results suggest that balancing should not be treated as a universally beneficial preprocessing step. Instead, its usefulness depends on the task type, imbalance severity, and model family. In highly imbalanced binary settings, especially in the CICIoMT2024 dataset, balancing improved the recognition of minority attack classes and often increased F1-Score and PR-AUC for tree-based ensemble models such as Random Forest and XGBoost.

In contrast, in datasets with already strong class separability, such as ECU-IoHT, balancing provided limited additional benefit. The results also indicate that deep learning models should be used more cautiously in multi-class balanced settings, where synthetic resampling may increase class overlap and reduce generalization. Based on these observations, we recommend first evaluating imbalance severity and class separability, then prioritizing balancing for highly imbalanced binary tasks and verifying its effect with class-sensitive metrics such as PR-AUC, FNR, and minority-class recall.

From an operational perspective, three factors should be checked before applying balancing: imbalance severity, task type, and feature separability. When the imbalance ratio is high and the task is binary, balancing is more likely to improve minority-class recognition, especially for tree-based ensemble models. When class separability is already strong, as observed in ECU-IoHT, the additional benefit of balancing may remain limited. In highly heterogeneous multi-class settings, particularly with many minority attack types, balancing should be applied more cautiously and should always be validated with class-sensitive metrics such as PR-AUC, FNR, and minority-class recall. Thus, the decision to balance should be based not only on class counts, but also on the interaction between dataset structure and model family.

Based on the imbalance ratios observed in this study, we interpret imbalance severity in a study-specific manner as follows: low imbalance for $IR < 5$, moderate imbalance for $5 \leq IR < 30$, high imbalance for $30 \leq IR < 100$, and extreme imbalance for $IR \geq 100$. These ranges should not be interpreted as universal thresholds, but as practical heuristics derived from the datasets examined here. In our experiments, balancing was generally unnecessary or only marginally beneficial under low imbalance with strong class separability, whereas it became more useful in highly imbalanced binary settings. In contrast, under extreme multiclass imbalance with many minority attack types, balancing required greater caution because synthetic resampling could improve minority-class recall while also increasing class overlap and weakening generalization.

The feature-importance profile in Figure 4 indicates that the strongest multiclass discrimination is driven primarily by timing-, rate-, and flag-related flow descriptors. This supports the interpretation that the best-performing classical models are exploiting structured network-behavior signatures, which may partly explain their stronger robustness relative to the current deep learning baselines on the same tabular feature space.

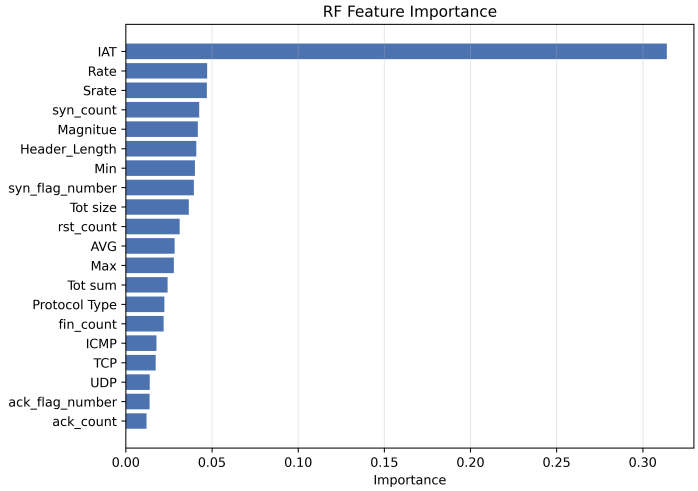


Figure 4. Top-20 feature importances for the RF model in the CICIoMT2024 19-class task under the balanced-training/original-testing scenario. The most influential features are dominated by flow timing, rate, and flag-related descriptors, indicating that multiclass discrimination is primarily driven by network-behavior signatures rather than by a single protocol field.

5. Conclusions

Internet of Medical Things (IoMT) devices play crucial roles in various healthcare services such as patient care, telemedicine, medication management, and health monitoring. These devices can improve treatment processes through continuous measurements, offer mobility and simplicity in hospital settings, and provide greater comfort to patients.

The security of IoMT devices is of critical importance because attacks on these devices can jeopardize patient privacy, data integrity, and service availability. Therefore, advanced techniques must be employed to enhance IoMT security.

Data balancing techniques are widely used to improve the performance of machine learning and deep learning models. However, in IoMT intrusion detection, their role extends beyond a preprocessing step. Because rare attack types are often underrepresented in network traffic datasets, models trained on imbalanced data may favor majority classes and produce false negatives for minority attacks. In healthcare environments, such false negatives have direct security implications, since undetected attacks may compromise patient privacy, alter medical data, or disrupt critical services.

Overall, the findings indicate that balancing is most useful in highly imbalanced intrusion-detection tasks when the baseline class distribution suppresses minority attack patterns, whereas its benefit becomes limited in datasets with high class separability or may even decrease for some deep learning models in multi-class settings. For this reason, balancing should be selected according to task type, imbalance severity, and model family, and should be validated using class-sensitive metrics rather than overall accuracy alone.

In this study, various data balancing methods were applied on open-source IoMT datasets: ECU-IoHT, WUSTL, and CICIoMT2024. These datasets represent different IoMT environments and attack scenarios. The WUSTL dataset was created using an advanced health monitoring system test environment and includes both network flow metrics and patient biometric data. The ECU-IoHT dataset was built on an IoHT environment where various attacks were carried out, designed to analyze attack behaviors and develop countermeasures. The CICIoMT2024 dataset aims to provide a realistic benchmark dataset and was generated by executing 18 different attacks in an IoMT test environment with 40 devices.

Balancing methods such as RandomUnderSampler, ADASYN, Borderline-SMOTE, and SMOTE were applied. For the ECU-IoHT dataset, it was observed that balancing procedures slightly improved accuracy in the binary (attack and normal) structure, with similar minor

improvements in the five-class structure. For the CICIoMT2024 dataset, in the binary structure, no single balancing method was consistently successful across all models, and in some cases, balancing did not yield a performance increase. Similarly, in the six-class and nineteen-class structures, balancing did not consistently lead to improved performance.

When considering the results from all datasets and models as a whole, it was observed that the combination of RandomUnderSampler and Borderline-SMOTE provided superior performance on original test data for models trained with balanced training sets. This method achieved the highest success in ten out of ten tests, with the K-Nearest Neighbors, Logistic Regression, AdaBoost, and XGBoost models each delivering the highest F1-Score in two tests, while Random Forest and DNN achieved high results in only one test. Next, RandomUnderSampler with ADASYN provided the highest success in seven results, with Logistic Regression and AdaBoost each achieving high F1-Scores in two tests, and XGBoost, CNN, and DNN in only one. RandomUnderSampler with SMOTE achieved the highest success in two tests with the Random Forest model.

In some cases, the original imbalanced structure of the dataset delivered superior performance. The performance of balancing methods varied depending on the dataset and model used. These findings demonstrate that, to ensure the security of IoMT devices and enhance the effectiveness of machine and deep learning models, data balancing strategies should be carefully selected and comprehensively tested.

For future work, the next logical step is to compare the foundational methods from this study against a broader range of state-of-the-art techniques. Such a study should evaluate advanced resampling methods that build upon the concepts in our work, including hybrid techniques like SMOTE-ENN and cluster-aware approaches like KMeans-SMOTE. Furthermore, it should benchmark the performance of cutting-edge generative models, such as CTAB-GAN+ and TabDDPM, directly against their higher computational costs. Finally, this comparison should include algorithm-level alternatives like Class-Balanced Loss to provide a complete picture. The results would offer a practical guide for developers on how to build IoMT attack detection systems that are both effective and efficient.

Author Contributions: E.G.: Writing—original draft preparation, software, formal analysis. B.U.: Writing—review and editing, methodology, formal analysis, conceptualization. G.U.: Writing—review and editing, methodology, supervision, formal analysis, conceptualization. I.S.: Writing—review and editing, methodology, supervision, formal analysis, conceptualization. All authors have read and agreed to the published version of the manuscript.

Funding: This work was supported by Office of Scientific Research Projects of Karadeniz Technical University. Project number: FBA-2026-17392.

Institutional Review Board Statement: Not applicable.

Informed Consent Statement: Not applicable.

Data Availability Statement: All datasets used in this study are publicly available benchmark datasets.

Conflicts of Interest: The authors declare no conflicts of interest.

References

1. Kumar, S.; Tiwari, P.; Zymbler, M. Internet of Things is a revolutionary approach for future technology enhancement: A review. *J. Big Data* **2019**, *6*, 111. [CrossRef]
2. Gubbi, J.; Buyya, R.; Marusic, S.; Palaniswami, M. Internet of Things (IoT): A vision, architectural elements, and future directions. *Future Gener. Comput. Syst.* **2013**, *29*, 1645–1660. [CrossRef]
3. Miorandi, D.; Sicari, S.; De Pellegrini, F.; Chlamtac, I. Internet of things: Vision, applications and research challenges. *Ad Hoc Netw.* **2012**, *10*, 1497–1516. [CrossRef]
4. Al-Fuqaha, A.; Guizani, M.; Mohammadi, M.; Aledhari, M.; Ayyash, M. Internet of things: A survey on enabling technologies, protocols, and applications. *IEEE Commun. Surv. Tutor.* **2015**, *17*, 2347–2376. [CrossRef]

5. Atzori, L.; Iera, A.; Morabito, G. The Internet of Things: A survey. *Comput. Netw.* **2010**, *54*, 2787–2805. [CrossRef]
6. Zanella, A.; Bui, N.; Castellani, A.; Vangelista, L.; Zorzi, M. Internet of Things for smart cities. *IEEE Internet Things J.* **2014**, *1*, 22–32. [CrossRef]
7. Whitmore, A.; Agarwal, A.; Da Xu, L. The Internet of Things—A survey of topics and trends. *Inf. Syst. Front.* **2015**, *17*, 261–274. [CrossRef]
8. Islam, S.R.; Kwak, D.; Kabir, M.H.; Hossain, M.; Kwak, K.S. The internet of things for health care: A comprehensive survey. *IEEE Access* **2015**, *3*, 678–708. [CrossRef]
9. Statista Research Department. Internet of Things (IoT) Statistics & Facts. 2003. Available online: https://www.statista.com/outlook/tmo/internet-of-things/worldwide (accessed on 15 November 2025).
10. Tawalbeh, L.; Muheidat, F.; Tawalbeh, M.; Quwaider, M. IoT privacy and security: Challenges and solutions. *Appl. Sci.* **2020**, *10*, 4102. [CrossRef]
11. Andy, S.; Rahardjo, B.; Hanindhito, B. Attack scenarios and security analysis of MQTT communication protocol in IoT system. In Proceedings of the 2017 4th International Conference on Electrical Engineering, Computer Science and Informatics (EECSI), Yogyakarta, Indonesia, 19–21 September 2017; pp. 1–6. [CrossRef]
12. Vaccari, I.; Aiello, M.; Cambiaso, E. SlowITe, a novel denial of service attack affecting MQTT. *Sensors* **2020**, *20*, 2932. [CrossRef]
13. Iqbal, W.; Abbas, H.; Daneshmand, M.; Rauf, B.; Bangash, Y.A. An in-depth analysis of IoT security requirements, challenges, and their countermeasures via software-defined security. *IEEE Internet Things J.* **2020**, *7*, 10250–10276. [CrossRef]
14. Sridevi, S.; Monica, K.M.; Prabha, R.; Senthil, G.A.; Narasimha Reddy, K.; Razmah, M. Network intrusion detection system using supervised learning based voting classifier. In Proceedings of the 2022 International Confonference of Communication, Computing and Internet of Things (IC3IoT), Chennai, India, 10–11 March 2022; pp. 1–6. [CrossRef]
15. Arshed, M.A.; Jabbar, M.A.; Liaquat, F.; Chaudhary, U.M.; Karim, D.; Alam, H.; Mumtaz, S. Machine learning with data balancing technique for IoT attack and anomalies detection. *Int. J. Innov. Sci. Technol.* **2022**, *4*, 490. [CrossRef]
16. Zhang, Y.; Liu, Q. On IoT intrusion detection based on data augmentation for enhancing learning on unbalanced samples. *Future Gener. Comput. Syst.* **2022**, *133*, 213–227. [CrossRef]
17. Farhan, B.I.; Jasim, A.D. Performance analysis of intrusion detection for deep learning model based on CSE-CIC-IDS2018 dataset. *Indones. J. Electr. Eng. Comput. Sci.* **2022**, *26*, 1165–1172. [CrossRef]
18. Dey, I.; Pratap, V. A comparative study of SMOTE, Borderline-SMOTE, and ADASYN oversampling techniques using different classifiers. In Proceedings of the 2023 3rd International Conference on Smart Data Intelligence (ICSMDI), Trichy, India, 30–31 March 2023. [CrossRef]
19. Dadkhah, R.; Neto, E.C.P.; Ferreira, R.; Molokwu, R.; Sadeghi, S.; Ghorbani, A.A. CICIoMT2024: A benchmark dataset for multi-protocol security assessment in IoMT. *Internet Things* **2024**, *28*, 101351. [CrossRef]
20. Batista, G.E.A.P.A.; Prati, R.C.; Monard, M.C. A study of the behavior of several methods for balancing machine learning training data. *SIGKDD Explor. Newsl.* **2004**, *6*, 20–29. [CrossRef]
21. Douzas, G.; Bacao, F.; Last, F. Improving imbalanced learning through a heuristic oversampling method based on k-means and SMOTE. *Inf. Sci.* **2018**, *465*, 1–20. [CrossRef]
22. Kotelnikov, A.; Baranchuk, D.; Rubachev, I.; Babenko, A. TabDDPM: Modelling tabular data with diffusion models. In *Proceedings of the 40th International Conference on Machine Learning (ICML 2023)*; JMLR: Norfolk, MA, USA, 2023; Volume 202, pp. 17717–17734.
23. Zhao, Z.; Kunar, A.; Birke, R.; Chen, L.Y. CTAB-GAN+: A cluster-based conditional table GAN. In *Proceedings of the 3rd ACM International Conference on AI Finance*; ACM: New York, NY, USA, 2022; pp. 37–45. [CrossRef]
24. Lin, T.-Y.; Goyal, P.; Girshick, R.; He, K.; Dollár, P. Focal loss for dense object detection. In *Proceedings of the 2017 IEEE International Conference on Computer Vision (ICCV)*; IEEE: New York, NY, USA, 2017; pp. 2999–3007. [CrossRef]
25. Cui, Y.; Jia, M.; Lin, T.-Y.; Song, Y.; Belongie, S. Class-balanced loss based on effective number of samples. In Proceedings of the 2019 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR), Long Beach, CA, USA, 15–20 June 2019; pp. 9268–9277. [CrossRef]
26. Li, X.; Zhao, H.; Xu, J.; Zhu, G.; Deng, W. APDPFL: Anti-poisoning attack decentralized privacy enhanced federated learning scheme for flight operation data sharing. *IEEE Trans. Wirel. Commun.* **2024**, *23*, 19098–19109. [CrossRef]
27. Yao, R.; Zhao, H.; Zhao, Z.; Guo, C.; Deng, W. Parallel convolutional transfer network for bearing fault diagnosis under varying operation states. *IEEE Trans. Instrum. Meas.* **2024**, *73*, 3540713. [CrossRef]
28. Zhao, H.; Liu, C.; Dang, X.; Xu, J.; Deng, W. Few-shot cross-domain fault diagnosis of transportation motor bearings using MAML-GA. *IEEE Trans. Transp. Electrif.* **2026**, *12*, 1165–1174. [CrossRef]
29. LeCun, Y.; Bengio, Y.; Hinton, G. Deep learning. *Nature* **2015**, *521*, 436–444. [CrossRef] [PubMed]
30. Jordan, M.I.; Mitchell, T.M. Machine learning: Trends, perspectives, and prospects. *Science* **2015**, *349*, 255–260. [CrossRef] [PubMed]
31. Goodfellow, I.; Bengio, Y.; Courville, A. *Deep Learning*; MIT Press: Cambridge, MA, USA, 2016.

32. Tavallaee, M.; Bagheri, E.; Lu, W.; Ghorbani, A.A. A detailed analysis of the KDD Cup 99 data set. In *Proceedings of the 2009 IEEE Symposium on Computational Intelligence for Security and Defense Applications, Ottawa, ON, Canada, 8–10 July 2009*; IEEE: New York, NY, USA, 2009; pp. 1–6.
33. Sharafaldin, I.; Lashkari, A.H.; Ghorbani, A.A. Toward generating a new intrusion detection dataset and intrusion traffic characterization. In *Proceedings of the 4th International Conference on Information Systems Security and Privacy (ICISSP), Funchal, Portugal, 22–24 January 2018*; Springer: Cham, Switzerland, 2018; pp. 108–116.
34. Sculley, D.; Holt, G.; Golovin, D.; Davydov, E.; Phillips, T.; Ebner, D.; Chaudhary, V.; Young, M. Hidden technical debt in machine learning systems. In *Advances in Neural Information Processing Systems*; NeurIPS: San Diago, CA, USA, 2015; Volume 28.
35. Chawla, N.V.; Bowyer, K.W.; Hall, L.O.; Kegelmeyer, W.P. SMOTE: Synthetic minority over-sampling technique. *J. Artif. Intell. Res.* **2002**, *16*, 321–357. [CrossRef]
36. Kuhn, M.; Johnson, K. *Applied Predictive Modeling*; Springer: New York, NY, USA, 2013.
37. Mittelstadt, B.D.; Allo, P.; Taddeo, M.; Wachter, S.; Floridi, L. The ethics of algorithms: Mapping the debate. *Big Data Soc.* **2016**, *3*, 2053951716679679. [CrossRef]
38. Boulle, M.; Huynh, T. OpenML: An R package to connect to the machine learning platform. *J. Mach. Learn. Res.* **2020**, *21*, 1–5.
39. WUSTL. WUSTL EHMS 2020 Dataset for Internet of Medical Things (IoMT) Cybersecurity Research. Dataset Webpage. 2020. Available online: https://www.cse.wustl.edu/~jain/ehms/index.html (accessed on 17 November 2025).
40. Ahmed, M.; Byreddy, S.; Nutakki, A.; Sikos, L.F.; Haskell-Dowland, P. *ECU-IoHT: A Dataset for Internet of Healthcare Things*; Edith Cowan University: Joondalup, Australia, 2021. [CrossRef]
41. He, H.; Garcia, E.A. Learning from imbalanced data. *IEEE Trans. Knowl. Data Eng.* **2009**, *21*, 1263–1284. [CrossRef]
42. Han, H.; Wang, W.Y.; Mao, B.H. Borderline-SMOTE: A new over-sampling method in imbalanced data sets learning. In *Advances in Intelligent Computing*; Springer: Berlin/Heidelberg, Germany, 2005; pp. 878–887. [CrossRef]
43. Bunkhumpornpat, C.; Sinapiromsaran, K.; Lursinsap, C. Safe-Level-SMOTE: Safe-Level-Synthetic Minority Over-Sampling Technique for handling the class imbalanced problem. In *Advances in Knowledge Discovery and Data Mining;* Lecture Notes in Computer Science; Springer: Berlin/Heidelberg, Germany, 2009; Volume 5476, pp. 475–482.
44. He, H.; Bai, Y.; Garcia, E.A.; Li, S. ADASYN: Adaptive synthetic sampling approach for imbalanced learning. In Proceedings of the 2008 IEEE International Joint Conference on Neural Networks, Hong Kong, China, 1–8 June 2008; pp. 1322–1328. [CrossRef]
45. Bishop, C.M. *Pattern Recognition and Machine Learning*; Springer: New York, NY, USA, 2006.
46. Freund, Y.; Schapire, R.E. A decision-theoretic generalization of on-line learning and an application to boosting. *J. Comput. Syst. Sci.* **1997**, *55*, 119–139. [CrossRef]
47. Guo, G.; Wang, H.; Bell, D.; Bi, Y.; Greer, K. KNN model-based approach in classification. In *On the Move to Meaningful Internet Systems 2003: CoopIS, DOA, and ODBASE*; Springer: Berlin/Heidelberg, Germany, 2003; pp. 986–996. [CrossRef]
48. Hosmer, D.W.; Lemeshow, S.; Sturdivant, R.X. *Applied Logistic Regression*, 3rd ed.; Wiley: Hoboken, NJ, USA, 2013. [CrossRef]
49. Breiman, L. Random forests. *Mach. Learn.* **2001**, *45*, 5–32. [CrossRef]
50. Chen, T.; Guestrin, C. XGBoost: A scalable tree boosting system. In *Proceedings of the 22nd ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*; ACM: New York, NY, USA, 2016; pp. 785–794. [CrossRef]
51. LeCun, Y.; Bottou, L.; Bengio, Y.; Haffner, P. Gradient-based learning applied to document recognition. *Proc. IEEE* **1998**, *86*, 2278–2324. [CrossRef]

Disclaimer/Publisher's Note: The statements, opinions and data contained in all publications are solely those of the individual author(s) and contributor(s) and not of MDPI and/or the editor(s). MDPI and/or the editor(s) disclaim responsibility for any injury to people or property resulting from any ideas, methods, instructions or products referred to in the content.