Computers, Materials & Continua

Tech Science Press

https://doi.org/10.32604/cmc.2026.081665

ARTICLE

Enhancing IoMT Network Threat Detection with Data Balancing for Multi-Class Attack Classification on CICIoMT2024 Dataset

Taghreed Alkhodaidi[1,*], Wadee Alhalabi[1] and Miada Almasre[2]

[1]Department of Computer Science, Faculty of Computing and Information Technology, King Abdulaziz University, Jeddah, Saudi Arabia

[2]Information Technology Department, Faculty of Computing and Information Technology, King Abdulaziz University, Jeddah, Saudi Arabia

*Corresponding Author: Taghreed Alkhodaidi. Email: talkhodaidi@stu.kau.edu.sa

Received: 06 March 2026; Accepted: 06 May 2026; Published: 15 June 2026

ABSTRACT: The rapid growth of the IoMT has resulted in critical security threats to healthcare infrastructure, which require highly sophisticated IDSs that can detect a wide range of and unbalanced attack patterns. This study has addressed a critical challenge faced by network security data, which is class imbalance, by presenting a comprehensive evaluation of data balancing techniques on both a real-world standard data set, CICIoMT2024, and a synthetic data set, SynIoMT2026, which we generated to mimic the characteristics of the standard data set for developing a highly controlled data set. Three data balancing techniques, ADASYN, Sample Weighting, and a hybrid technique involving both SMOTE and SMOTEEN, were systematically applied and evaluated on the severely class-imbalanced data set, wherein the majority classes, such as DDoS_UDP with ~2M instances, far outweigh the minority classes, such as Recon_Ping_Sweep with 926 instances. The balanced data sets were used to train and evaluate a range of ML models, including random forest, AdaBoost, logistic regression, and DNN models for binary classification, 6-class classification, and 19-class classification. The proposed method achieved outstanding results, with 99.8% model accuracy achieved for binary classification. The results of the evaluation have demonstrated the robustness of the random forest algorithm, which showed accuracy ranging from 97% to 99% in all scenarios. The results have demonstrated the potential of strategic balancing in unlocking the potential of the model, especially in the results obtained from the AdaBoost model, where the SMOTE-SMOTEEN technique showed a significant increase in accuracy in 6-class classification from 69.8% to 91.6%, and even more dramatic results in 19-class classification, increasing accuracy from 23.6% to 51.9%. This has demonstrated the need to select the optimal balancing technique to unlock the potential of the model. The results have also demonstrated high accuracy in the SynIoMT2026 synthetic dataset, showing 99% accuracy in training, covering six categories, and 89% accuracy in nineteen categories, with minimal overhead. This study has demonstrated the viability of using synthetic datasets in model development and has provided a balanced dataset that has been tested in both real-world and synthetic environments.

KEYWORDS: IoMT; SMOTE; data balancing; cybersecurity; machine learning; ADASYN; sample weighting; IDS

1 Introduction

In the modern world, where digital transformation is omnipresent, the importance of cybersecurity has reached new heights, especially since organizations are increasingly dependent on technology for their smooth functioning and growth. The emergence of new-age attacks on infrastructure highlights the necessity for innovative cybersecurity strategies that can effectively address such attacks. Cybersecurity refers to

various techniques and instruments used for protecting electronic devices such as computers, servers, networks, data centers, and information systems against various attacks, threats, vulnerabilities, etc. Attacks on electronic devices often cause disruptions in their functioning, which may even cause the complete deletion of data on the system. Cybersecurity, therefore, focuses on identifying such attacks, responding accordingly, and restoring the system.

The emergence of artificial intelligence (AI) heralds a new era of technological advancement for various industries, which has completely changed the way businesses operate, thereby changing traditional ways of doing business. The emergence of AI has significant implications for industries such as healthcare, finance, transportation, etc., which has led to unparalleled levels of efficiency, innovation, and analysis. The move towards a digital world facilitated by AI, however, has brought with it significant challenges, one of which is the increasing complexities of cyber threats, which often involve AI for attacking security systems, exploiting vulnerabilities, etc. This phenomenon presents evidence of the dualistic nature of technological progress, in which progress in technology can facilitate nefarious activities [1]. As cyber attacks escalate in terms of complexity and frequency, it has become increasingly evident that conventional cybersecurity measures like antivirus software, intrusion detection systems (IDS), and firewalls have become ineffective in addressing cyber attacks [2].

The innovative solution to this problem has been provided by AI, which has been found to be extremely effective in addressing the problem of cybersecurity [3]. The creation of AI-based algorithms to learn from past cyber attack patterns has been found to be extremely important in creating effective cybersecurity measures [3]. The integration of AI with cybersecurity measures has been found to be extremely effective in addressing cybersecurity issues through AI-based strategies that utilize methodologies like machine learning (ML), deep learning (DL), natural language processing (NLP), and anomaly detection [4]. AI-based strategies have been found to have the capability to learn from vast data sets and to quickly respond to sophisticated cyber attacks.

Medical devices, sensors, and systems all work within the IoMT, generating vast amounts of extremely sensitive information. This interconnectedness fundamentally changes the way in which medical information is gathered, processed, and used for a variety of purposes, including sophisticated diagnostic tools, health monitoring, etc. Improved health outcomes, tailored treatment regimens, constant surveillance, etc., are all examples of how these new medical technologies can significantly improve health outcomes. The increased interconnectedness of these IoMT devices, however, also brings about a variety of new security concerns. To protect this information from unauthorized access, breaches, etc., which all become critical concerns as the interconnectedness of these IoMT devices grows, strong security measures must be put in place to protect the complex and sensitive information gathered by IoMT devices [5].

Table 1 lists a variety of IoMT devices, including their description, cybersecurity vulnerabilities, etc. Every device, from glucose monitors to smart infusion pumps, plays a vital role in health care, but each also has its own unique cybersecurity vulnerabilities. For instance, continuous glucose monitors are vulnerable to interception and manipulation, while wearable cardiac devices are vulnerable to hijacking, denial-of-service (DDoS) attacks, etc.

Table 1: Cybersecurity relevance of various IoMT devices.

IoMT Device	Description	Cybersecurity Relevance
Glucose Monitors CGMs	Devices like Dexcom or FreeStyle Libre track blood glucose levels and transmit data wirelessly.	Vulnerable to data interception, spoofing, and manipulation.

(Continued)

Table 1 (continued)

IoMT Device	Description	Cybersecurity Relevance
Blood Pressure Monitors	Connected cuffs that send data to apps or healthcare providers.	Risk of unauthorized data access or fake data injection.
Wearable Cardiac Devices	Devices like Zio Patch or portable ECG devices monitor heart conditions remotely.	Critical vulnerabilities: interception, device hijacking, DoS attacks.
Smart Infusion Pumps	Pumps that deliver medication doses (e.g., insulin) and can be configured remotely.	Target of ransomware, configuration tampering, and remote control risks.
Smart Inhalers	Connected devices that monitor inhaler use for asthma and COPD patients.	Exposed to attacks affecting medication adherence monitoring.
Remote Patient Monitoring Systems	Systems that collect patient vital signs and send them to healthcare providers.	Data privacy breaches, MITM (Man-in-the-Middle) attacks.
Implantable Cardiac Devices	Implanted devices that can be remotely monitored.	Life-threatening risks if unauthorized commands are sent to the device.
Smart Thermometers	Connected thermometers providing continuous temperature data.	Data integrity issues, especially during pandemics.
Mobile Medical Apps	Apps that aggregate data from multiple medical wearables (blood pressure, oxygen, glucose).	Application layer vulnerabilities (e.g., poor authentication).
Hospital Asset Tracking Sensors	RFID and IoT tags for tracking medical equipment and supplies.	Target for location spoofing, theft, or data exfiltration.

In order to improve the security of the network, IoMT organizations have been using private networks along with redundant backup devices, which is an extension of the general security measures that have been put in place for IoT networks, including encryption, authentication of the devices, segmentation, anomaly detection through machine learning, etc., [6]. However, the management of communication systems for the integration of the data has become significantly more complex due to the exponential growth of the medical information, which is also becoming more heterogeneous. This is also emphasized by the recent advancements made in the diversity-aware frameworks, including the use of proactive caching for mobile networks [7]. As such, the conventional measures have failed to satisfy the changing security requirements of IoMT organizations. This is an issue of critical significance, considering the interrelation between the safety, health, and security of the medical devices, which is of broader implications. For the achievement of adequate network security, the management of the data in an efficient manner in real-time is of critical significance for the confidentiality of the information [8].

The integration of ML into IoMT security is considered a paradigm shift from traditional signature-based detection systems towards intelligent and adaptable systems that have the capability to identify new

and dynamic cyber attacks [9–11]. By analyzing huge amounts of network traffic and device behavior data, ML systems have the ability to learn complex patterns that can be considered malicious, ranging from DDoS floods to small-scale reconnaissance attacks. This is considered vital for protecting sensitive healthcare infrastructure, as the impact of an attack is not limited to data compromise but can be life-threatening. Therefore, developing effective ML-based IDS systems has emerged as an important research priority, which is considered vital for protecting the confidentiality, integrity, and availability of medical devices and critical data they handle [11].

A dataset is considered an aggregate of data that is used to train, validate, and test ML systems [12]. This is considered the fundamental component upon which all ML processes are built, which can be considered the textbook from which the algorithm learns. The performance of an ML system is considered directly dependent upon the quality, quantity, and relevance of data upon which it is trained. A flawed dataset will always produce a flawed model.

1.1 The Challenge of Class Imbalance in Cybersecurity Dataset

The overall efficacy of the machine learning models is primarily based on the quality and nature of the datasets provided during the training phase. The real-world datasets, including the ones associated with the security of IoMT, often show class imbalance and varying states of the data, leading to biased results and high accuracy rates that show poor performance when the models are subjected to the critical minority classes. Thus, this research focuses on the application of advanced data preprocessing techniques to overcome the limitations of the models, ensuring the efficacy of the threat detection models.

1.1.1 Balanced Dataset

A balanced dataset is defined as a dataset in which the classes, usually labeled as positive and negative classes, are represented in approximately equivalent proportions. This equilibrium reduces the natural bias that machine learning models are prone to when they favor the majority class, thus improving their ability to generalize to both common and rare classes.

1.1.2 Imbalanced Dataset

Generally, the characteristics of an imbalanced dataset are the presence of a large gap between the two classes, with one of the classes (the majority or the negative class) far outweighing the other (the minority or the positive class). An example of an imbalanced dataset is in fraud detection, where the number of fraudulent transactions is very low compared to the number of non-fraudulent transactions.

To solve the problem of a data imbalance, there are several resampling strategies that can be used, and these are:

- Random Under-Sampling (RUS): In this technique, the majority class is reduced randomly to the level of the minority class. Even though this technique is effective in balancing the data, it is also possible to lose some potentially valuable information in the majority class.
- Random Over-Sampling (ROS): In this technique, the minority class is expanded by copying the instances in the minority class to the level of the majority class. Though this technique is simple and works well, there is a possibility of overfitting.
- Cluster-Based Over-Sampling: In this technique, the data is clustered in the majority and minority classes, and the instances are oversampled in each cluster to the level where the instances are evenly distributed in the majority and minority classes.

- Synthetic Minority Over-Sampling Technique (SMOTE): Instead of replicating minority class instances, SMOTE creates synthetic samples by interpolating between instances, thereby improving diversity among minority class instances and preventing overfitting caused by duplicated data.
- Adaptive Synthetic Sampling (ADASYN): This approach is an extension of SMOTE, which creates synthetic data for minority class instances, especially those that are difficult to learn, by assigning a weighting factor to each minority instance based on local density. This approach is best used with complex data sets that have irregular decision boundaries.
- SMOTE with Edited Nearest Neighbors (SMOTE-ENN): This approach is a combination of SMOTE and ENN, which is used to clean the data set created by SMOTE. This approach has two components: SMOTE is first used to over-sample the minority class, and then ENN is applied to both classes, removing instances misclassified by their three nearest neighbors.
- Sample Weighting (SW): This is an algorithm-level approach, which is incorporated into an algorithm by assigning a higher misclassification cost to minority class instances, thereby forcing the algorithm to focus on correctly classifying minority instances without changing the structure of the data set.

1.2 Motivation

The efficacy of the anomaly detection systems based on machine learning is heavily dependent on the quality and representativeness of the datasets used in the system. However, the existing network security datasets are also characterized by a number of limitations, which are notable in terms of the diversity of the devices, communication protocols, and types of attacks. Such limitations are particularly notable in the context of the Internet of Medical Things (IoMT), where the unique characteristics of the medical devices and the healthcare environments are also reflected in the network environments, creating a number of unique threat conditions. In the context of the existing datasets, it is also notable that the inherent imbalance and non-uniformity, which are characteristic of the network environments in the context of the Internet of Medical Things, are not well represented in the datasets, where some types of attacks are exceedingly rare, while some other types of attacks are relatively common. In this context, there is a pressing need to develop comprehensive datasets that are able to represent the irregular nature of the IoMT in a realistic manner, considering the inherent imbalance and non-uniformity of the network environments. In this context, the research on the problem of imbalance in the datasets and the importance of the balanced representation in the context of the learning models is of paramount importance.

This research work presents an extensive empirical assessment of the effectiveness of data balancing methods for mitigating class imbalance in the context of IoMT-based multi-class attack classification. The effectiveness of various data balancing methods is extensively evaluated for standard datasets. The results are significant in the context of creating robust intrusion detection systems in IoMT-based healthcare environments by highlighting the significance of data balancing in achieving accurate multi-class threat detection.

The paper is organized as follows: Section 2 reviews current related works on securing the IoMT and the associated challenges in attack detection. Section 3 outlines the proposed methodology. Section 4 presents the datasets. Section 5 presents the methodology for generating the SynIoMT2026 dataset and modeling the attack signature. Section 6 discusses the results and performance. Finally, Section 7 concludes the study.

2 Related Works

The availability of datasets for the IOMT is still limited. This is why research focused on IoT is more prominent, as both IoT and IoMT are vulnerable to similar types of attacks, such as distributed DDoS and spoofing attacks. For example, research using the CICIoT2022 dataset has shown promising results,

indicating that various ML models achieve high accuracy rates, such as Decision Tree at 98.5%, AdaBoost at 98.7%, XGBoost at 95.6%, and KNN at 98.6% [13].

The research by Ramesh et al. [14] offers two major contributions to intrusion detection in IoMT-based systems. First, it addresses the essential problem of model size optimization, particularly for dealing with large-scale datasets, by proposing efficient architectures optimized for the storage and processing constraints of IoMT devices in the real world. Second, the research utilizes an extensive evaluation approach, including robust K-fold cross-validation, focusing on F1 scores, to ensure the accurate evaluation of the proposed models, particularly for dealing with imbalanced datasets.

In this regard, this study relies on recent advancements made in network security, as cited in [15], which employed various ML models to assess their effectiveness for IDS in IoMT networks. The study employed an ensemble model based on XGBoost, which improves performance for particular types of attacks. In addition, it employed sequential models, such as the Long Short-Term Memory (LSTM) network and the CNN-LSTM model, which take into account temporal dependencies in data. Finally, it employed unsupervised models, such as Autoencoders and Isolation Forest, which have been proven to be effective for anomaly detection. The results showed high performance for some of these models. Specifically, it is worth noting that both the Isolation Forest and Autoencoder models showed high effectiveness, with precision, recall, and F1-scores higher than 0.90 for multiple protocols. In addition, it is worth noting that the performance of the one-class Support Vector Machine (SVM) is reliable, with precision = 0.91 and recall = 0.90, although its F1-score is slightly lower, at 0.83, implying that it is highly effective for normal traffic but slightly less effective for other types of anomalies. Finally, regarding the ensemble models, it is worth noting that the Ensemble Stacking model, which relies on Logistic Regression as its meta-learner, showed the highest performance, achieving an accuracy of 0.96 for cross-validation, which is higher than all other models. A novel federated learning framework has been proposed by Misbah et al. that is particularly designed to meet the security and computation-related limitations of the IoMT. In this study [5], the limitations of the centralized model are addressed using advanced FL techniques, federated dynamic averaging, and stacking, to develop a distributed model of attack classification in the edge devices. The study is validated using the CICIoMT2024 dataset. Among the models used in the study, the ensemble models, particularly the Random Forest model, have shown promising results, with an accuracy of 99.22% [5].

Alsbatin et al. also used the CICIoMT2024 dataset without preprocessing and balancing the data [16]. In this study, the results showed that the Random Forest model is significantly better in the detection of cyberattacks in the IoMT environment, with an accuracy of 94.8% and a reliability rate of 96.1%. Thus, the Random Forest model is the best model in the detection of cyberattacks in the IoMT environment, providing higher robustness and reliability compared to other models used in the study [16].

A resource-efficient IDS is proposed by Salehpour et al. [17] specifically for the Internet of Medical IoMT environment. In the proposed IDS, a two-stage feature selection method is used. In the first stage, the features are selected using the mutual information filtering method. In the second stage, the features are ranked using ensemble-based feature selection techniques, such as Random Forest, AdaBoost, XGBoost, and LightGBM. For the final classification, a Random Forest classifier is used. In the proposed IDS, the results show a significant increase in accuracy, precision, and F1-score when the system is evaluated on the three popular datasets, namely, WUSTL-EHMS-2020, NSL-KDD, and CICIoMT2024, particularly in the detection of DDoS and DOS attacks.

The importance of domain-specific IoMT datasets and preprocessing techniques in the development of robust IDS is highlighted in the study by Doménech et al. [18]. In the study, the performance of machine learning models is evaluated on a general IoT dataset, CICIoT2023, and an IoMT dataset, CICIoMT2024. The results show a significant effect of the domain on the performance of the IDS model, with a considerable drop

in F1-score, up to 66.87%, when the model is evaluated on the other dataset. In the study, the design of the CICIoMT2024 dataset is critically evaluated, and several baseline optimization techniques are proposed, such as uniform windowing, structured data splitting, temporal dependency correction in time series analysis, and advanced dataset balancing techniques. In the study, the implementation of these techniques results in a considerable improvement in the overall IDS performance, with a model accuracy of 0.9985.

The research study by Mohsin and Jony [19] utilizes data balancing techniques with the Synthetic Minority Oversampling Technique (SMOTE) and other advanced ML models for investigating cyber threats targeting IoMT devices. The research aims to improve the security of healthcare systems by identifying and mitigating cyberattacks more effectively. The authors perform extensive experiments to identify the best ML models for three types of classification granularities, namely binary, multiclass, and multilevel classifications of cyberattacks. AdaBoost, Random Forest, k-Nearest Neighbors (kNN), and XGBoost are some of the techniques that show promising results for accurate classifications of several types of attacks, validating their usage for IoMT device security applications.

Class imbalance is a major problem associated with ML, and it affects the overall evaluation and applicability of ML-based security detection models. In the research study on malware detection using API call data, Goyal and Kumar [20] performed a comparative evaluation of the ML model's performance on imbalanced and balanced datasets. The results show that the Random Forest technique demonstrated a significant difference in the accuracy rates of the imbalanced and balanced datasets, i.e., 98.94% and 90.38%, respectively. Critically, the authors observe that the improved accuracy of the model on the imbalanced dataset might be misleading since the improvement could be the result of overfitting the majority class, compared to the more reliable and generalizable performance of the model on the balanced dataset. This observation underscores the need for addressing the issue of class imbalance in ML-based security systems prior to the training of the model in order to avoid misleading performance metrics and improve the overall practicality of ML-based security systems. The study is in tandem with the consensus in the ML literature that dataset balancing is critical in the development of ML-based classifiers, especially in the context of cybersecurity applications.

In the traditional approach to addressing the issue of class imbalance in ML-based security systems, undersampling of the majority class and/or oversampling of the minority class are employed in order to balance the dataset prior to the training of the ML model. However, the issue of class imbalance in ML-based security systems has traditionally been overshadowed by the need to develop more effective learning algorithms. However, the need for the integration of balance correction and data pruning mechanisms into the learning process has come to the fore. This study [21] provides an extensive overview of conventional and state-of-the-art solutions for handling class imbalance by employing intelligent sampling strategies for majority and minority classes before training the model. In addition, it covers an extensive review of hybrid sampling strategies, which focus on retaining difficult-to-learn instances, which are challenging for classification accuracy, and eliminate easy-to-learn instances, which provide little learning value. The state-of-the-art techniques have been designed to improve learning efficiency by allocating maximum learning resources to instances that provide maximum learning value, thereby improving classification accuracy on imbalanced datasets without compromising vital minority-class information. This study [22] emphasizes the significance of handling class imbalance, which is vital for machine learning (ML) models. This study presents basic balancing techniques and provides an extensive review of imbalanced learning solutions under three categories: data-level, algorithm-level, and hybrid-level solutions. The study has been conducted to assess the effectiveness of these solutions to handle bias towards majority classes and improve minority-class instance detection. Shah Mirkhail and Zhang [23] have developed an innovative architecture for the detection of intrusions in IoMT systems, based on the combination of an Autoencoder (AE) for anomaly detection

and an LSTM network for the detection of patterns. Such an approach is theoretically sound, leveraging the capabilities of the two approaches. Moreover, the evaluation that the authors have conducted, based on the complete CICIoMT2024 dataset, is an important strength, which enhances the credibility of the results. Despite the fact that the accuracy achieved, which is 94.1%, is high, the paper does not provide enough information for the evaluation of the results, based on the comparison with other contemporary approaches, such as other types of deep learning architectures, which limits the assessment of the actual advantage that the architecture under review presents. More seriously, the paper discusses the problem of class imbalance, which is significant, and the fact that the authors have employed oversampling and undersampling techniques. However, the paper does not discuss the actual methods that have been employed, which is an important methodological limitation. Such an approach may significantly affect the accuracy of the results, inflating the actual performance of the architecture, which may focus more on the detection of the majority class, such as the DDoS attack, without actually improving the detection rates for other types of attacks, which may even be decreased. Such an approach does not actually reflect the actual capabilities of the architecture under review.

Although, as stated, Areia et al. [24] make a valuable contribution with the IoMT-TrafficData dataset, as it directly addresses the notable problem of scarcity of IoMT-related data for IDS-related research, a critical evaluation of the paper reveals some notable weaknesses, limiting its potential for transformation. First and foremost, the scope of the dataset, with only eight types of attacks, is rather limited and may not reflect the full scope of IoMT-related vulnerabilities, as well as possible attacks on certain medical device-related protocols or even patient-related data. Additionally, although the results show high performance with an F1-score of more than 90%, this should also be contextualized, as the results are intrinsically connected with the authors' dataset composition and balance, and may not reflect the results in noisier and imbalanced network environments, as may be encountered in the wild. On the other hand, the results, as they relate to the superiority of flow-based features over packet-based ones, may be considered rather obvious, as this is a more holistic approach, as may be expected in network-related security issues. Overall, although this paper is a valuable contribution, as it is a necessary tool, it is also considered an initial step and will depend on the ability of the research community to validate and test the results with more diverse and adversarial threat scenarios.

The research undertaken by Rehman et al. [25] offers a pragmatic and timely review of feature selection in the context of machine learning-based intrusion detection for IoMT networks, highlighting a vital concern related to the inherently resource-constrained nature of medical devices. The application of filter-based techniques such as Fisher Score, Mutual Information, and Information Gain for intrusion detection using two benchmark datasets, namely CICIoMT2024 and IoMT-TrafficData, offers valuable insights of significant practical utility. One of the main findings of the research, namely the fact that a remarkably small number of features, i.e., 3–4, is sufficient for optimum binary classification, is both a strength and a weakness of the research, as such a small number of features offers significant advantages in terms of reduced complexity and improved efficiency, although it is still unclear whether such a small number of features is sufficient for optimum performance in the context of complex and adaptive types of attacks, as well as noisier and more realistic clinical settings. Similarly, the finding related to the application of Information Gain with XGBoost offering excellent results is of some value, although such results are also in line with conventional best practice, and the identification of key feature categories is of some utility for practical purposes, although such a finding is also insufficient for offering deeper insights into the reasons why such features are more discriminative for IoMT-specific types of threats, as well as offering a more mechanistic understanding of the results obtained with such techniques.

Yazdinejad et al. [26] note that security operations center (SOC) analysts suffer significant performance degradation under elevated cognitive stress, yet existing systems treat stress detection and decision support as separate problems. To bridge this gap, they propose a cognitive–physiological synchronization (CPS) framework for IoT-based SOCs (IoT-SOCs) that integrates a calibrated DNN-XGBoost ensemble for multimodal stress inference from ECG, EDA, and respiration signals. Their CPS layer converts physiological beliefs into cognitive utilities via Bayesian log-odds updates, dynamically aligning decisions with the analyst's real-time stress state. They further introduce a utility-aware temporal reasoner (UATR) for smoothing sequential evidence and a stress-weighted memory (SWM) mechanism for adaptive experience recall within the SpeedyIBL cognitive model. Evaluated on the WESAD dataset (leave-one-subject-out cross-validation), the framework achieves 95.8% accuracy (AUC = 0.967) with subsecond latency, and in zero-shot CICIDS2017 simulations, it improves decision stability and reduces false escalations compared to rule-based baselines Yazdinejad et al. [26]. These results demonstrate that synchronizing physiological stress inference with cognitive policy selection enhances end-to-end action quality under uncertainty.

Wahab et al. [27] address the challenge of DDoS attacks in IoT networks, which can severely disrupt communication and cause operational losses. They propose a deep learning-based intrusion detection system using three architectures—CNNs, DNNs, and Transformer-based models—on the CICIoT2023 dataset with log normalization and SMOTE oversampling. Their DNN achieves 99.2% accuracy for binary classification, near-perfect performance (99.9%–100%) for three-class classification (benign, DDoS, non-DDoS), and 93.0% accuracy for 12-class classification, outperforming state-of-the-art methods in detection accuracy and efficiency. These results demonstrate the potential of integrating advanced DL models into IDS frameworks for scalable and effective IoT network security.

AlFuraih et al. [28] address the need for accurate and interpretable intrusion detection systems for IoT networks facing evolving cyberattacks. They propose an explainable hybrid CNN–XGBoost framework for multi-class IoT attack classification using the CICIoT-DIAD 2024 dataset. Their pipeline employs scalable chunk-wise preprocessing, Random Forest-based top-k feature selection, and a comparison between leakage-prone and leakage-aware feature ranking strategies to reduce selection bias. A one-dimensional CNN learns a 128-dimensional representation from the selected features, followed by XGBoost for final multi-class classification. Under the leakage-aware protocol, the model achieves 0.9324 accuracy with a macro-F1 of 0.5910, demonstrating that leakage-aware selection provides more defensible generalization estimates while maintaining competitive detection performance. Additionally, they apply SHAP to interpret model decisions in the latent space, revealing that only a small number of embedding dimensions contribute most of the decision evidence—a finding that can aid analyst triage, though the explanations remain indirect with respect to original traffic features. This work highlights the importance of both detection accuracy and interpretability in IoT IDS design.

In recent study [29], the authors proposed a novel Tsetlin Machine (TM)-based Intrusion Detection System for detecting a wide range of IoMT cyberattacks. Unlike traditional black-box approaches, the TM is a rule-based, interpretable machine learning method that models attack patterns using propositional logic. Extensive experiments on the CICIoMT2024 dataset, which includes multiple IoMT protocols and attack types, demonstrate that the TM-based IDS achieves 99.5% accuracy for binary classification and 90.7% for multi-class classification, surpassing existing state-of-the-art methods. Furthermore, to enhance model trust and interpretability, the framework provides class-wise vote scores and clause activation heatmaps, offering clear insights into the most influential clauses and dominant classes contributing to final decisions. This work underscores the value of interpretable AI for IoMT security, balancing high detection performance with model transparency.

Challenges in IoMT Intrusion Detection Research

Several works have been conducted on the application of ML for intrusion detection in general IoT networks, whereas research focused on the IoMT ecosystem is surprisingly scarce. The communication characteristics, protocol usage (notably MQTT for medical data communication), and operation-specific security needs of medical devices are quite different from those of regular IoT devices such as smart home appliances. The majority of the models are trained and tested on general IoT datasets (notably KDDCup99 [16], NSL-KDD [17], and CICIDS2017 [30]), which cannot encapsulate the threat characteristics of medical device networks. Hence, a generalization gap is observed, where the models may perform well for general IoT networks but may perform poorly or may not be dependable for IoMT networks.

Further, though the problem of class imbalance is recognized as a major problem in the application of ML, its impact on IoMT security is often neglected or insufficiently addressed. The CICIoMT2024 dataset, similar to other intrusion detection datasets, is heavily imbalanced, with some classes (notably DDoS_UDP) having a significantly higher number of instances compared to other classes (notably Recon_Ping_Sweep). Numerous research works report high accuracy results, with the results being potentially skewed due to the imbalanced nature of the dataset or apply oversampling techniques such as Random Oversampling without benchmarking the results with other oversampling techniques such as SMOTE, ADASYN, and SMOTEENN, and algorithmic techniques such as class weighting for multi-class IoMT attack classification and prediction.

A significant part of the studies conducted in this domain have shown theoretical models with high performance metrics, but limited opportunities for practical verification. There is also a lack of transition from static documentation to dynamic, accessible, and verifiable research. However, the paper also identifies the need for the results to be explained, along with the provision of an actual platform that helps in the replication of the research with ease. The paper addresses the actual problem of the lack of verifiability of the results through the use of the CICIoMT2024 dataset for the benchmarking of various balancing techniques for the classification of attacks into multiple classes, along with the provision of an interactive platform for verifiability. This paper takes an actual step towards the creation of strong, reliable, and applicable security solutions for the infrastructure of the healthcare system.

3 Methodology

To address the significant class imbalance problem associated with the CICIoMT2024 dataset, a variety of data balancing techniques was employed. This was necessary to ensure that the optimal data balancing technique was used to overcome the model bias associated with the majority class of attacks, thus improving the overall detection of cyberattacks. The dataset was balanced using the following techniques: ADASYN, SMOTE, SMOTE-ENN, and SW. These techniques were used individually, and their effectiveness was determined by training the machine learning model on the balanced dataset, thus providing a fair comparison of the effectiveness of these techniques. The inclusion of ADASYN and SMOTE-ENN techniques in this study is necessary to conduct a critical analysis of the effectiveness of these techniques in addressing the class imbalance problem associated with the IoMT dataset, thus improving the overall model performance in detecting cyberattacks.

The model was trained to perform binary classification, distinguishing between normal and attack traffic. The model was later extended to perform 6-class classification, including the following classes: Benign, DDoS, MQTT_Attack, DoS, Reconnaissance, and Spoofing. Finally, the model was modified to perform multi-class classification, including eighteen different classes of attacks. To evaluate the model, four algorithms were previously evaluated on the real-world IoMT dataset in [31], enabling direct comparison between real and synthetic data performance. Specifically:

- Logistic Regression—interpretable baseline, low computational cost (suitable for edge IoMT devices).
- Random Forest—handles mixed numeric/categorical features, robust to outliers in sensor data.
- AdaBoost—effective for imbalanced attack classes common in IoMT.
- Deep Neural Network—captures complex non-linear patterns in multivariate network traffic flows. Accuracy, precision, recall, and F1 score metrics were calculated individually using these models.

All balancing techniques and the four algorithms for evaluation were independently applied to each of the two datasets, and then the performance metrics were compared systematically. As illustrated in Fig. 1, the methodology defines the steps involved in the multilevel classification analysis.

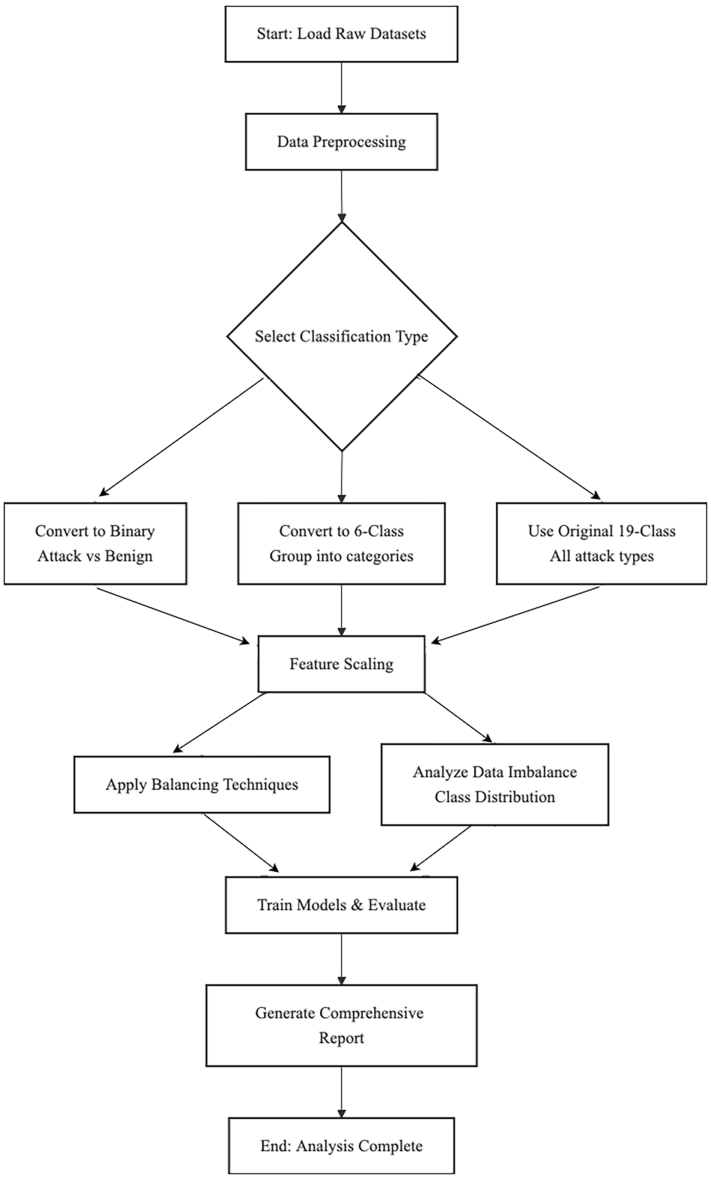

Figure 1: Comprehensive methodology for multi-level classification analysis.

4 Datasets

This section will discuss the CICIoMT2024 and Synth2026 datasets. The Synth2026 dataset is a synthetic dataset designed to mimic the CICIoMT2024 dataset, allowing for the evaluation of the potential of synthetic data to cover the wide range of devices that are normally hard to access. The characteristics of the CICIoMT2024 and Synth2026 datasets will be discussed in the following sections.

4.1 CICIoMT2024

This study utilizes the CICIoMT2024 Dataset [31], created by the Canadian Institute for Cybersecurity (CIC) at the University of New Brunswick. This dataset was created to simulate a real-world IoMT network environment, including both malicious and benign traffic captures. This dataset includes packet captures (pcap) files, network flow information, and feature sets, thus aiding in the detection of cyber attacks in the IoMT ecosystem. This dataset differs from traditional datasets, such as the KDD Cup '99 Dataset or NSL-KDD, in that it includes multi-protocol attacks, thus making it more applicable to the current IoMT cybersecurity scenario.

The dataset archive has two directories, namely 'Bluetooth' and 'WiFi_and_MQTT.' The 'WiFi_and_MQTT' directory has 'attacks' and 'profiling' directories within it. Further, the 'attacks' directory has 'train' and 'test' directories containing training and testing data, respectively. These directories contain multiple CSV files, each containing different types of attacks. To analyze this dataset, these files must be combined, and the dataset must be described in terms of features and data types. Moreover, each file in the dataset is missing a label, thus requiring the addition of a label.

The dataset is a representation of a wide range of cyber threats, including distributed DDoS attacks, Man-in-the-Middle attacks, ransomware, botnets, network reconnaissance, and data exfiltration. All these attacks were conducted in a controlled environment using actual IoMT devices, which are exposed to adversarial traffic. The dataset includes multi-protocol traffic generated by 40 medical devices, with 25 devices being actual devices and 15 devices simulated. In addition, the dataset has 45 features and 18 different types of attacks, ranging from DDoS floods using TCP, ICMP, SYN, and UDP packets, as well as DDoS attacks, network reconnaissance, MQTT protocol-based exploits, and ARP spoofing. Furthermore, the dataset offers a comprehensive device profile, where the entire device lifecycle, from power-on to power-off, is captured, including device power states, device operation, and user interactions.

One notable aspect of the data collection environment is that it uses a Raspberry Pi, which is placed inside the lab as a malicious device to launch all attacks. It is important to note that the data was used for training without applying any balancing technique. In the data preprocessing step, network traffic data from the pcap files is converted into a structured feature set. Initially, dataframes do not have a label column, which is then added manually to label each data point as normal or an attack. In the feature extraction step from network traffic data, various important features were extracted, such as the magnitude of traffic flow vectors, covariance between various protocol volume levels, and radius of data clusters in feature space. In addition, network protocols such as IGMP for group management, ICMP for control messages, Telnet for remote login, SMTP for mail transport, SSH for secure login, and DHCP for dynamic IP address allocation were quantified. The dataset has 7,160,831 data points, each with 45 features. These features include protocol features, time features, flags, and statistical features. There is no explicit feature for attack patterns; instead, they are learned based on flags such as SYN, RST, and ACK flags, as well as Rate and Duration fields.

A correlation heatmap, as illustrated in Fig. 2, was employed to investigate the linear correlation between network traffic features, ranging from −1 (perfect negative correlation) to +1 (perfect positive correlation). First and foremost, the correlation heatmap is used to understand the structure of the feature space before

model training, since it reveals redundant features that exist in the form of clusters of highly correlated variables (marked in dark red with correlation coefficients close to +1). The problem of highly correlated features lies in their overlapping meaning, which leads to the problems mentioned above and results in multicollinearity and increased computation time, not to mention overfitting. Knowing that certain features provide redundant information enables us to reduce the dimensionality by either selecting one feature from the pair of highly correlated variables or extracting features using algorithms like PCA. The second use of a correlation heatmap involves detecting features that show almost zero correlation (in white or light blue color) with other features, which is crucial since these are the features providing independent information on traffic behavior. Thus, keeping them will enhance generalizability and increase model accuracy when dealing with unknown types of attacks. Lastly, it should be noted that there are many features showing negative correlation (dark blue color in the chart). They may be extremely useful for the purpose of intrusion detection, since the inverse relationship of two features may indicate an anomaly in the network traffic.

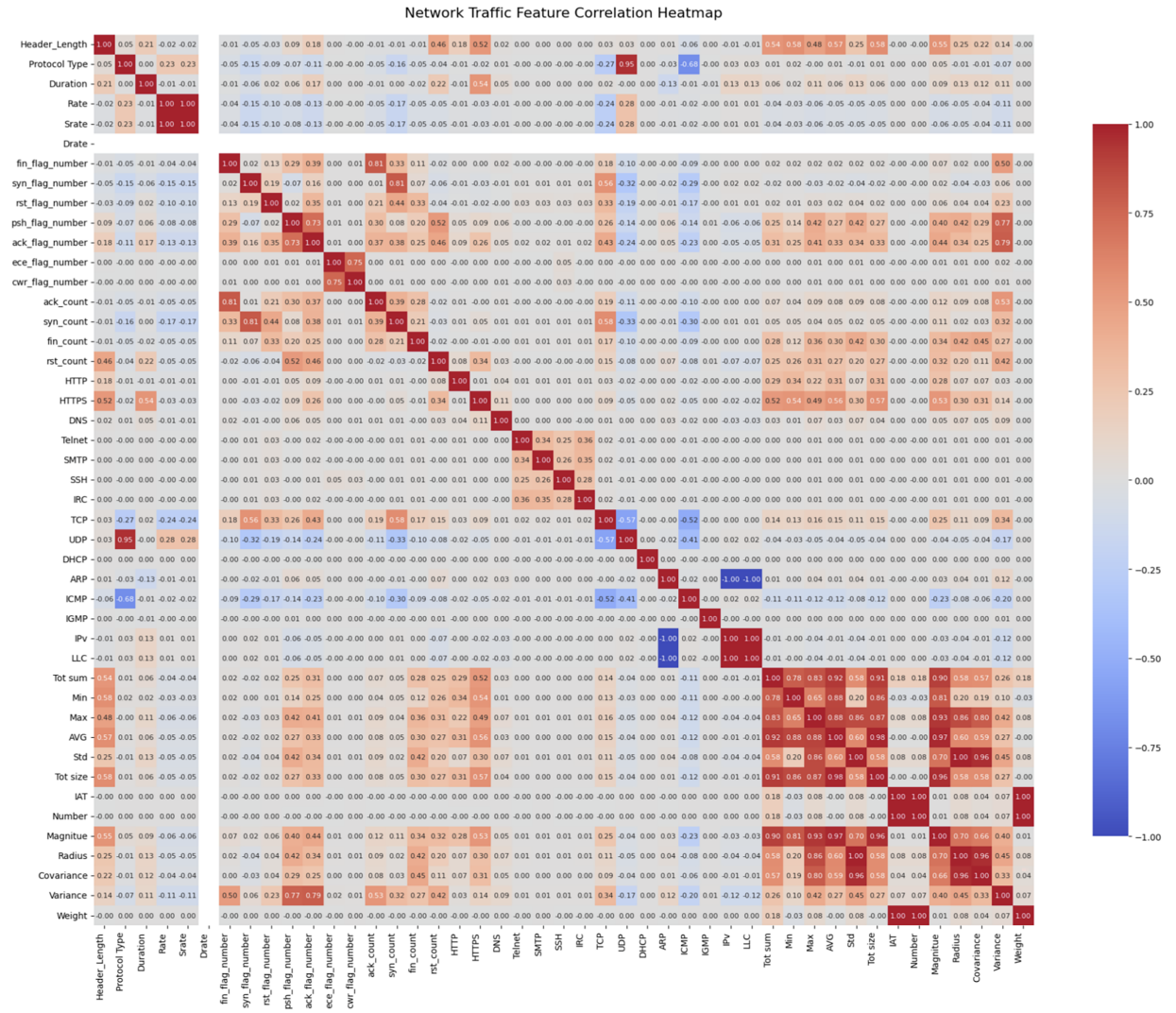

Figure 2: Correlation heatmap of CICIoMT2024 dataset.

The matrix demonstrates a pattern of near-perfect correlation along the dominant diagonal of the matrix, with clusters of highly correlated features (as highlighted in red), in addition to areas of low

correlation (as depicted in lighter shades of color or blue). These strong correlations between the features indicate the possibility of redundancy in the information being conveyed, which might be addressed in order to improve the efficiency of the model. On the other hand, the presence of features with low correlation values, near zero (as depicted in white or light blue), is highly desirable since they are likely to represent unique aspects of the network operations. The presence of negative correlations (as depicted in blue) further indicates the possibility of distinguishing certain types of attacks from benign operations, as they represent opposite traffic flows. This analysis has confirmed the presence of diverse features in the set, which is highly desirable for developing an efficient intrusion detection model.

The feature importance chart in Fig. 3 provides key information about what network traffic parameters have the biggest impact on IoMT intrusion detection, thus serving our two research objectives of optimizing the model as well as making it more interpretable. Firstly, the feature importance chart provides us with quantitative insights into the role each particular feature plays in prediction, allowing for efficient data-driven feature selection. Thus, high importance features (IAT = 0.21, Rate = 0.05, AVG = 0.04) have been found to be the best predictors of an attack and should therefore be left in the final model. On the other hand, low or even negative importance scores (dominant_protocol = −0.07, syn_ack_ratio = −0.06, Variance = −0.05) indicate the lack of correlation between the feature and the dependent variable and should therefore be discarded, improving the performance, reducing the amount of time required for training, as well as preventing potential overfitting issues. Secondly, the chart helps us interpret the results obtained, providing clear indications not only of the overall accuracy but also of the particular network behavior that differentiates normal data traffic from attacks. For the security experts, it is important to know IAT and packet rate to be able to monitor suspicious activities. Finally, it confirms the efficiency of our preprocessing techniques such as duplication removal (5119 duplicated values were removed) and SMOTENC-based class balancing.

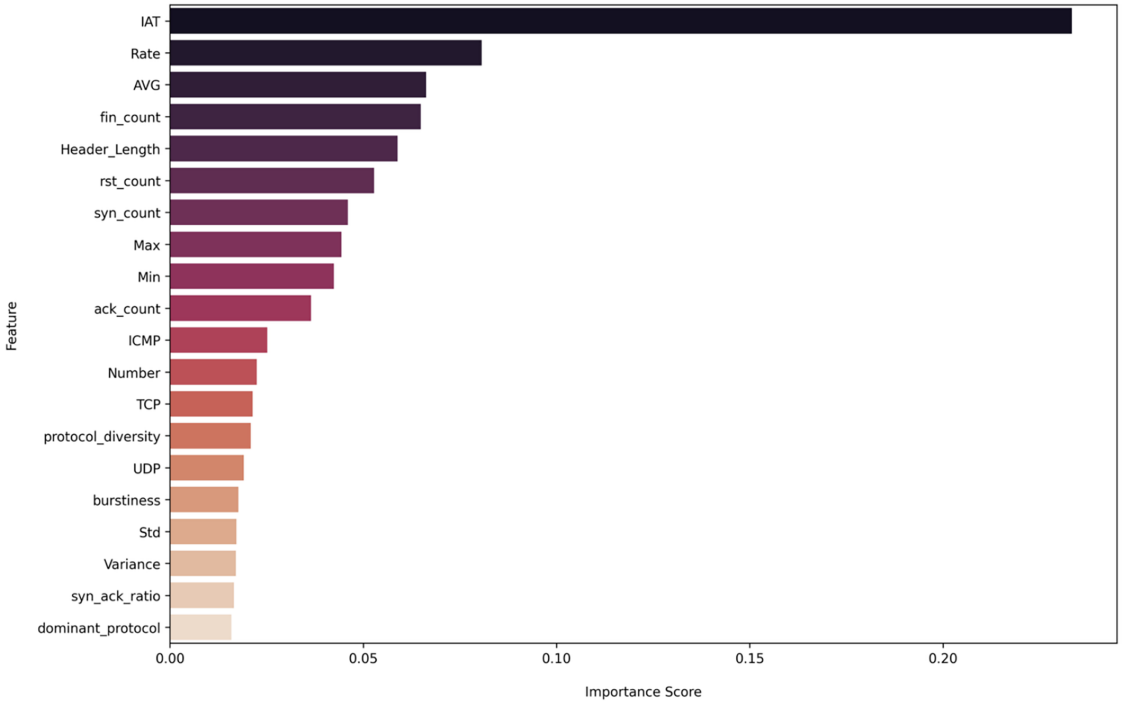

Figure 3: Top 20 network traffic features for attack detection for CICIoMT2024.

The class distribution was significantly imbalanced before the balancing step, with the majority of the data belonging to a single type of attack, DDoS_UDP, with ~1.9 million samples. The other classes were significantly underrepresented, e.g., Recon_Ping_Sweep had only 926 samples, while MQTT_Malformed had even fewer samples. The benign traffic belonged to the minority class with 230,339 samples, as illustrated in Fig. 4. After the balancing step, all classes become equal, as illustrated in Fig. 5.

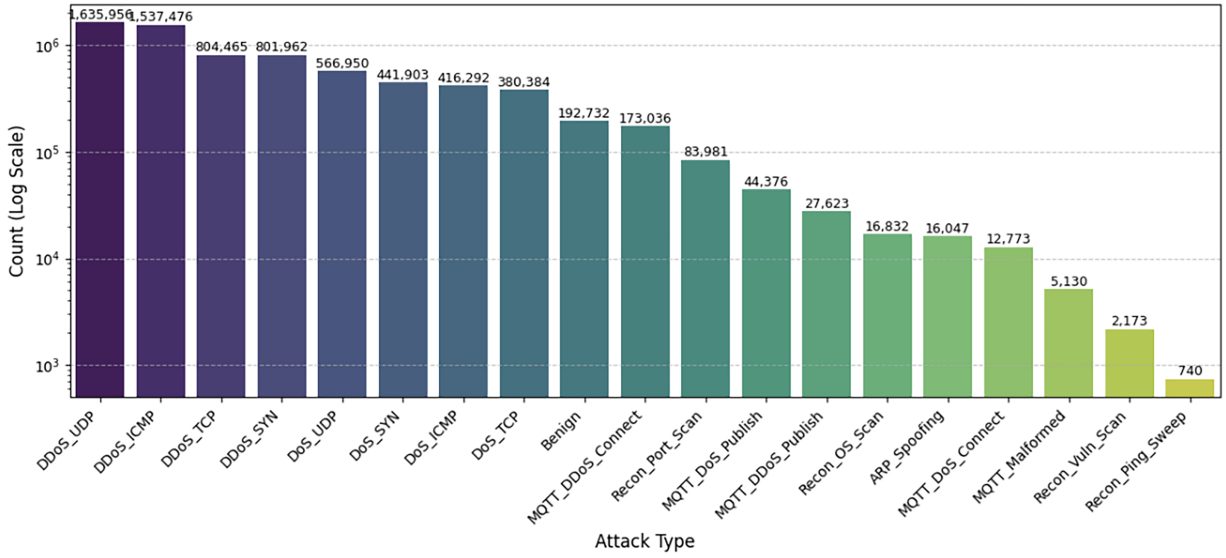

Figure 4: Original attacks distribution for CICIoMT2024.

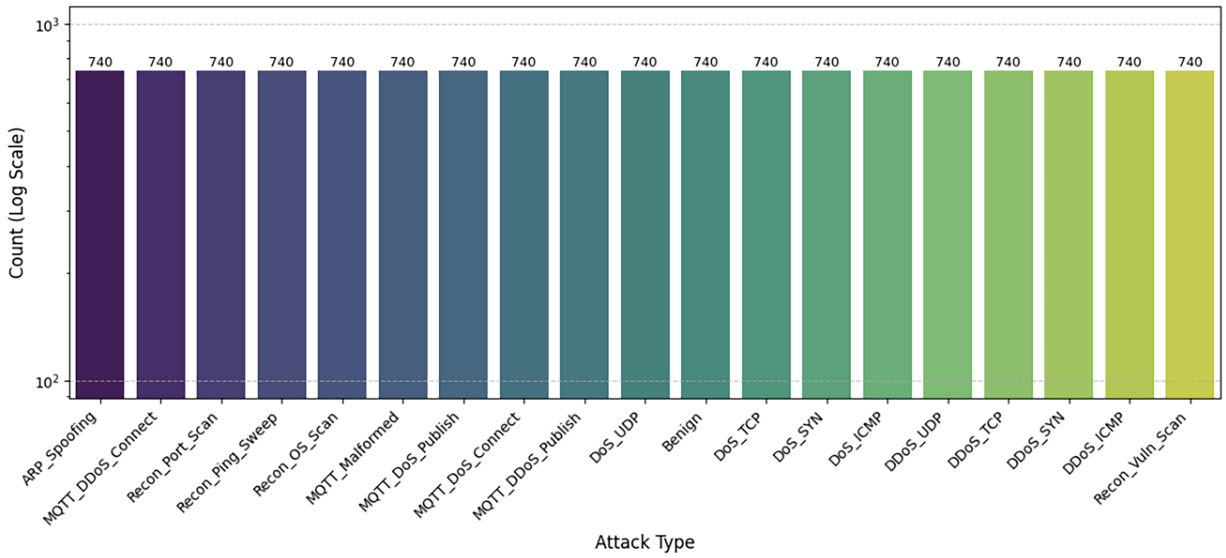

Figure 5: Balanced attack distribution for CICIoMT2024.

4.2 SynIoMT2026

In order to make experimentation in IoMT security flexible and reproducible, an interactive web application was developed, which is also capable of generating the dataset. As shown in Fig. 6, the application was developed and hosted using the Replit environment. This application was useful in the generation of the

SynIoMT2026 dataset. The dataset was developed using Python and was rule-based, designed to mimic the statistical and networking aspects of the CICIoMT2024 benchmark. The application allows researchers to generate customized datasets according to their needs by defining the parameters of the dataset, which are as follows:

1. Device and Attack Scope: This parameter allows users to select from the basic set of simulated medical devices and 37 unique types of attacks, which include DDoS, reconnaissance, spoofing, and protocol-specific attacks.
2. Dataset Scale and Composition: This parameter allows users to define the size of the dataset and the Benign Ratio (normal traffic ratio).
3. Class Distribution: This parameter allows users to select the option of generating either balanced or imbalanced classes.

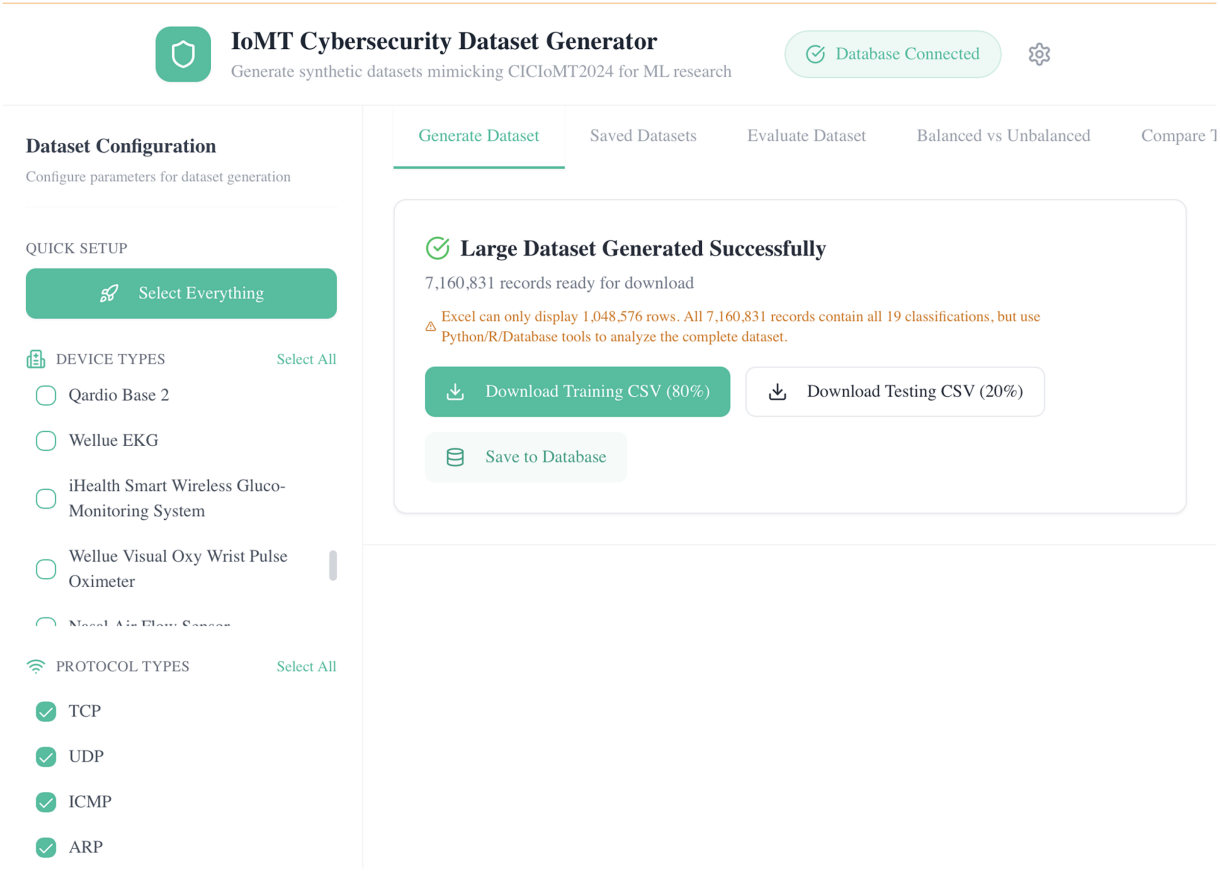

Figure 6: SynIoMT2026 dataset generator platform.

This approach allows for the controlled evaluation of the importance of the balancing of the dataset before training the model. The key contribution of the application is the ability to simulate multiple types of medical devices, which are usually hard to simulate in the real world because of the risks involved in conducting experiments in the medical field.

In addition to the above features, the application also includes an evaluation module that allows users to benchmark their machine learning models and compare the effectiveness of different balancing techniques to find the best one for achieving the highest accuracy in IoMT scenarios.

The creation of the SynIoMT2026 corpus is based on a rule-based approach, which aims to mimic the statistical and behavioral characteristics of realistic IoMT network traffic, in strict conformance with the CICIoMT2024 benchmark specification. The feature set was deliberately crafted to identify the discriminative characteristics of a wide range of cyber threats, enabling machine learning models to learn on semantically relevant patterns. The synthesis configuration is organized around several main dimensions.

Basic Distinctions between Benign and Malicious Activities

- Traffic Rate and Volume: Benign traffic is characterized by moderate and steady traffic volumes, with approximately 69 packets per second, based on the PCAP analysis. Also, the number of packets per flow is steady and predictable, with approximately 338 packets per flow. This is due to normal device communication, which includes sensor readings, status updates, and normal data communication. Malicious traffic, on the other hand, is characterized by significantly higher traffic volumes. For instance, the DDoS_UDP traffic has approximately 519 packets per second, the DDoS_TCP traffic has approximately 281 packets per second, and the Recon_Ping_Sweep traffic has approximately 139 packets per second.
- Packet Size Characteristics: The benign flows show a large distribution of packet sizes, ranging from a minimum of about 42 bytes to a maximum of about 1514 bytes, with an average of about 340 bytes. This is a large distribution, showing a variety of legitimate activities, including small acknowledgments and larger data transfers. Malicious traffic, on the other hand, has a number of distinctive packet size characteristics. To begin with, DDoS_ICMP has uniformly small packets, averaging about 60 bytes, while ARP_Spoofing averages about 482 bytes.
- Protocol Distribution: The benign traffic has a varied protocol distribution, with about 81% using TCP, while about 19% use UDP, as derived from the analysis of the PCAP file. Malicious traffic, however, has a number of distinctive protocol characteristics. To begin with, DDoS_UDP is purely UDP, using protocol number 17, while DDoS_ICMP is purely ICMP, using protocol number 1. MQTT-based attacks, on the other hand, use protocol number 6, which is TCP, while using MQTT ports. Finally, ARP_Spoofing has both TCP and UDP, but with the ARP flag active.
- TCP Flag Patterns: The benign traffic has a balanced distribution of TCP flags, including SYN, ACK, FIN, PSH, etc., reflecting a normal TCP three-way handshake, including a normal teardown.

The attacks show abnormal patterns:

- DDoS_SYN/DoS_SYN: 99.99 flags set to SYN, indicating that all connections are requested but never completed.
- DDoS_TCP/DoS_TCP: 0x0000 flags, indicating abnormal packets that do not adhere to normal TCP protocols.
- Recon_Port_Scan: High levels of SYN flags (0.63) and RST flags (0.32), which is normal for classical port scanning, where refused connections send resets.
- MQTT_Malformed: Combination of PSH and ACK flags, indicating malformed MQTT packets.
- Inter-Arrival Time (IAT): Benign traffic will normally have varied IAT, which is usually related to human interaction patterns or periodic sensor reporting patterns. In contrast, attacks will show reduced IAT, usually related to maximum transmission rates for floods and periodic reporting for scans.
- Flow Duration: Benign traffic will show moderate flow duration, usually related to normal communication patterns. In contrast, attacks will show varied flow duration:
- DDoS attacks will show longer duration with high intensity.
- Reconnaissance attacks, such as Port_Scan and OS_Scan, will show short duration as they usually involve short bursts.

- Weight Feature: Benign traffic will show low weights, normally related to normal communication priority levels. In contrast, attacks will show higher weights, normally related to higher priority levels, with each type of attack showing unique weight ranges:
- ARP_Spoofing will show weights ranging between 218 and 244, which is significantly different from benign traffic.

The above characteristics demonstrate the significance of the CICIoMT2024 dataset for machine learning. Each type of attack will have its unique combination of features, which is referred to as its "signature." Most important is that none of these features alone will be sufficient to differentiate all attacks from benign traffic; instead, it is the correlation between features, such as rate, flags, protocol, packet size, etc., that will enable accurate classification of all 19 types of attacks. This is important because it validates the significance of 46 features, which is important for accurate classification, instead of fewer features, which might have been considered sufficient.

The distribution of the SynIoMT2026 dataset is characterized by its long-tailed, highly imbalanced nature, which accurately reflects the nature of the IoMT network environment. As shown in the logarithmic distribution in Fig. 7, the number of attacks varies over several orders of magnitude. As shown in the logarithmic distribution in Fig. 7, volumetric attacks are the most predominant type of attack, followed by the identification of DDOS_SYN floods, which have the highest number of occurrences, followed closely by DDOS_TCP floods, each with over 200,000 attacks. As expected, volumetric attacks, particularly the two types mentioned, remain the most predominant type of attack in the IoMT network environment, as they have the least complexity yet the greatest disruptive power. Moving towards the center of the distribution, we find reconnaissance attacks such as Recon_Ping_Sweep and Recon_OS_Scan, which have moderate occurrence rates, reflecting the background noise that is characteristic of network scanning, which is commonly encountered in the IoMT network environment. Most importantly, the Benign class, which refers to normal, legitimate network traffic, has been carefully calibrated to reflect the nature of the IoMT network environment, where benign traffic is the majority class, yet in this dataset, we have included the entire spectrum of attacks targeting the IoMT network environment. Moving towards the tail end of the distribution, we find sophisticated, yet less occurring, attacks, including MQTT attacks such as MQTT_DoS_Connect, MQTT_DDoS_Publish, as well as reconnaissance attacks such as Recon_Vuln_Scan. As shown, the repeated occurrence of the attacks in the IoMT network environment is an attempt to elicit the occurrence of rare, yet impactful, attacks. As shown, the IoMT network environment is characterized by its highly imbalanced nature, which is one of the greatest challenges for IDSs, which must achieve high detection rates for the minority class attacks without being biased towards the overwhelming number of volumetric attacks, as well as the high number of benign attacks, while at the same time achieving low false positive rates.

As shown in the correlation heatmap provided in Fig. 8, a number of structural patterns are evident, which reinforce the semantic correctness of the generated network traffic features. Specifically, the flow-level features such as Tot_sum, Min, AVG, Max, Std, and Tot_size are strongly correlated, with a correlation coefficient ranging from 0.54 to 1.00, thus confirming the correct preservation of internally consistent mathematical relationships, which would be expected in a real-world network flow scenario. Moreover, the features Magnitude, Radius, Covariance, and Variance are also strongly correlated, ranging from 0.48 to 1.00, thus confirming the correct semantic representation of these features, which are derived from composite statistical measures of the flow characteristics.

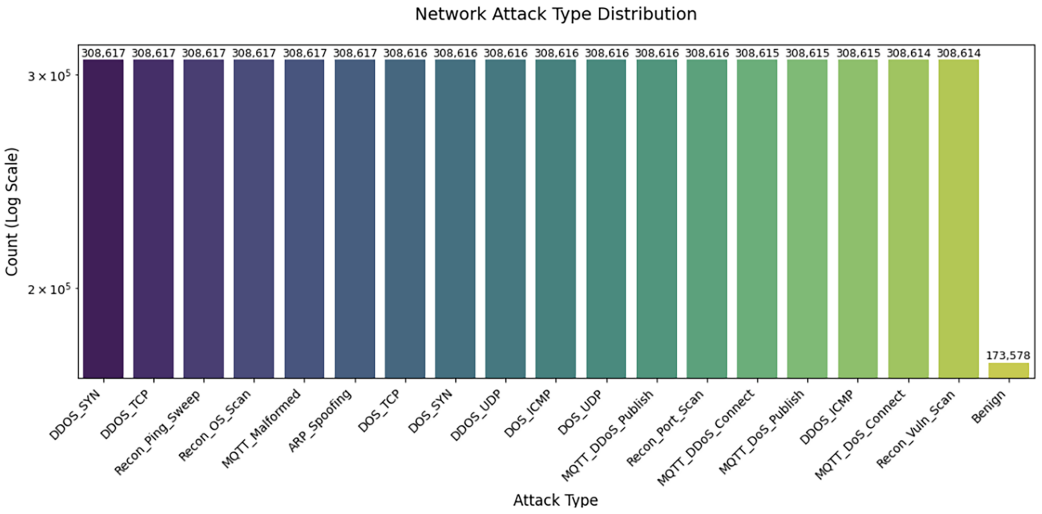

Figure 7: Attack type distribution for the SynIoMT2026 dataset.

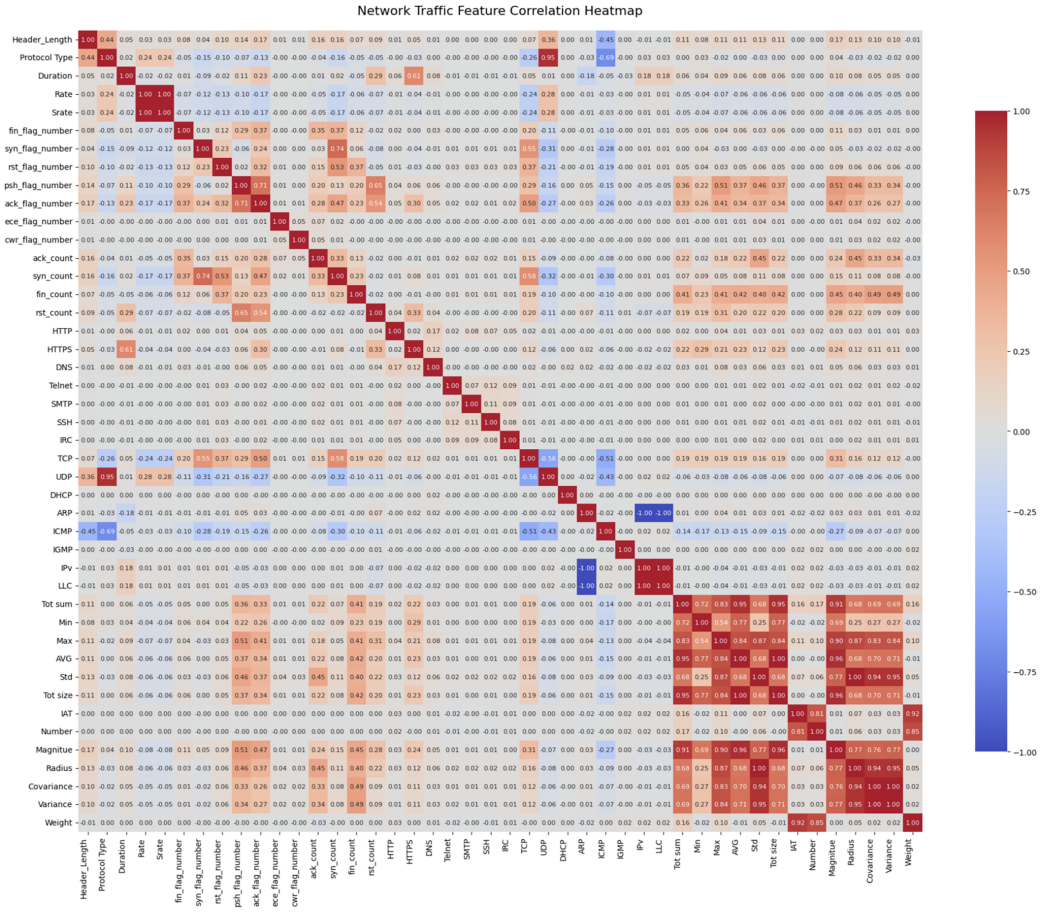

Figure 8: The correlation heatmap for the SynIoMT2026 dataset.

As shown in Fig. 8, the features associated with the TCP flag features block also demonstrate strong semantic consistency. Specifically, the pair of features ack_flag_number, ack_count has a strong positive correlation of 0.62, while the pair of features syn_count, syn_flag_number also has a strong positive correlation of 0.22, thus confirming the correct semantic representation of these features, which would be expected in a real-world TCP protocol scenario. Moreover, the feature ece_flag_number has a strong positive correlation with both syn_count (0.41) and the HTTP/HTTPS/DNS features block (0.54, 0.54), thus confirming the correct semantic representation of this feature, which would be expected in a real-world protocol scenario, reflecting the relationship between ECN-capable TCP protocol negotiation and application-layer protocol stacks used in modern protocol stacks.

As shown in Fig. 8, the protocol features block, comprising features such as TCP, UDP, DHCP, ICMP, IPv, has demonstrated the expected level of mutual exclusivity. Specifically, the pair of features TCP, UDP has a negative correlation of −0.47, thus confirming the correct semantic representation of these features, which would be expected in a real-world protocol scenario, since a flow would be associated with at most one protocol or another. Moreover, the pair of features Srate, Drate has a strong positive correlation of 1.00, while the feature Rate has a strong positive correlation of 0.37 with this pair, thus confirming the correct semantic representation of these features, which would be expected in a real-world protocol scenario, since a flow would be associated with a bidirectional.

The near zero correlations obtained across a number of feature blocks, including Telnet, SMTP, SSH, IRC, IGMP, and LLC, which indicate a lack of significant correlation with other features, suggest that these protocol features are sparse, independent, and only present in specific scenarios. The Weight feature is still largely uncorrelated with the majority of the network features, indicating that it is used independently to assist classification. Overall, the heatmap shows that the synthetic dataset has maintained the underlying correlations of a real-world IoMT network traffic dataset, including strong correlations where physical and protocol relationships are necessary, exclusive correlations where protocol features conflict, and weak correlations where features are semantically unrelated.

The feature importance analysis provided in Fig. 9 shows that the temporal and volumetric characteristics of the network traffic constitute the most discriminative features for the detection of IoMT attack types. The feature IAT is at the first position with a high importance score of 0.066, followed by the ‘Duration’ feature at the second position with an importance score of 0.061. This shows that the temporal behavior of the network traffic, i.e., the rate at which packets arrive at the network devices and the duration of the network sessions, is the most discriminative feature for the classification of the various attack types from the normal traffic. This is in conformity with the basic principle that the automated tools used for carrying out the attacks will have a significantly different temporal behavior compared to the normal communications between the various IoMT devices.

The ‘ICMP Protocol Indicator’ is at the third position with an importance score of 0.048, showing the significant differences between the ICMP-based DDoS, DoS, and Recon Ping Sweep attack types, which can easily be distinguished from the other traffic categories based solely on the protocol indicators. The ‘ece_flag_number’ (0.045), ‘traffic_intensity’ (0.045), and ‘Variance’ (0.045) features also show almost similar levels of importance, suggesting the significance of the behavior of the TCP protocols during the negotiations and the statistical variation in the characteristics of the network traffic for the classification purposes. The ‘Rate’ (0.044), ‘protocol_diversity’ (0.043), and ‘Drate’ (0.043) features also show the significance of the volumetric characteristics of the network traffic, i.e., the rate at which the traffic is flowing through the network and the protocols used for the communications, for the classification purposes.

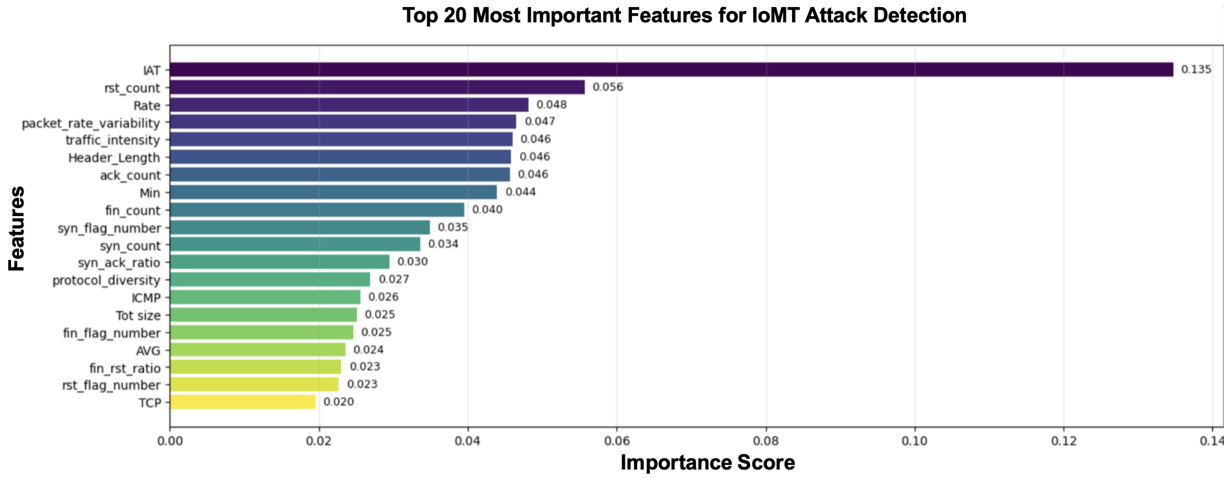

Figure 9: Top 20 network traffic features for attack detection for the SynIoMT2026 dataset.

The 'packet_rate_variability' (0.040), 'Covariance' (0.037), 'DHCP' (0.036), 'Max' (0.032), and 'ack_count' (0.032) features show the significance of the mid-tier features, which include the stability of the network traffic flows, the presence of the DHCP protocols, the maximum packet sizes, and the 'ack_count' flags, for the classification purposes. The presence of the 'syn_count' (0.031), 'IPv' (0.030), 'TCP' (0.028), 'Number' (0.025), 'rst_count' (0.024), and 'syn_flag_number' (0.024) features in the list of the top 20 features shows the significance of the behavior of the TCP protocols during the handshake mechanisms for the classification purposes.

ML models detect compound multi-feature signatures rather than relying on individual metrics, which is the claim for the Syn IoMT 2026 dataset. Each of the 19 classification types has a distinct combination of protocol type, flag pattern, packet sizes, rates, and timing. As an example, the Recon_Port_Scan type is detected by the presence of a SYN bias of approximately 0.63, the presence of a RST value of 0.32, short IAT, and a rate of approximately 144 packets/second. On the other hand, the DDoS_UDP type is detected by the presence of the highest rate of ~519 packets/second, the exclusive use of the UDP protocol, and the absence of inter-packet time delay.

4.3 Redundancy Mitigation

Three approaches were used to handle redundancy in the dataset. The first one involved discarding duplicate rows through df.drop_duplicates() to avoid pseudo-replication and any potential correlation inflation between features. The second approach was to remove redundant mathematical features (Magnitude, Radius, Variance, Covariance, Tot sum, Tot size, Weight, and Std), which were deterministic redundancies relative to the base measurement features. Lastly, PCA was conducted on the remaining standardized features to decorrelate variables by converting them into uncorrelated components and retaining at least 95% of the variance in the process. These methods helped reduce the number of features from 28 down to about 8–10 principal components.

4.4 SynIoMT2026 Dataset Validation

In order to prove the validity of the generated synthetic dataset, this section validates SynIoMT2026 by evaluating whether the generated synthetic dataset maintains the statistical and behavioral characteristics of actual IoMT data from CICIoMT2024. We validated SynIoMT2026 using two complementary statistical approaches.

4.4.1 Kolmogorov-Smirnov (KS) Feature-Wise Analysis

We performed two-sample KS tests on 44 common features using 100,000 samples from each dataset to compare marginal distributions. The results are represented in Table 2.

Table 2: Kolmogorov-smirnov test results for synthetic dataset (Result 2).

Metric	Value
Average KS statistic	0.101
Median KS statistic	0.023
Features with KS < 0.05	13/44 (30%)
Features with KS < 0.10	20/44 (45%)
Features with KS < 0.20	30/44 (68%)

The average value for KS statistics was observed to be 0.101, while its median value was observed to be 0.023. The proportion of KS < 0.10 was observed to be 45%, while that of KS < 0.20 was observed to be 68%. Protocol-level features including TCP, UDP, ICMP, ARP, DNS, HTTP, DHCP, and IGMP demonstrated strong agreement, with KS statistics being KS < 0.001.

4.4.2 PCA-Based Multivariate Overlap Analysis

In order to analyze the preservation of the joint correlation structure in the created dataset, principal component analysis was performed on 28 non-constant variables. The measure of similarity was calculated with the use of Jensen-Shannon divergence as per the methodology of Lin [32]. As can be seen from Table 3, a PCA overlap score of 96.6/100 was recorded by the generated dataset, which was associated with a Jensen-Shannon divergence of 0.034. This demonstrates high similarity between the multivariate distributions of the real dataset and the generated one. The top two principal components were responsible for 16.31% and 10.74%, respectively.

Table 3: PCA overlap results for synthetic dataset (Result 2).

Metric	Result
PCA Overlap Score	96.6/100
Jensen-Shannon Divergence	0.034
PC1 Explained Variance	16.31%
PC2 Explained Variance	10.74%
Total Explained Variance (PC1 + PC2)	27.05%
Features Used	28

With the PCA overlap of 96.6/100 (scores > 80 indicate excellent overlap), we can deduce that the generated synthetic dataset SynIoMT2026 retains the correlation of the original dataset CICIoMT2024 with regards to multivariate analysis. Moreover, with the low Jensen-Shannon divergence of 0.034, we know that there is high similarity between the two datasets in their latent structure, which can be contributed to by the elimination of duplicate rows and mathematical functions from the former dataset.

5 SynIoMT2026 Dataset Generation Methodology and Attack Signature Modeling

For the development of the model, the dataset was split into the training set and test set, with an approximate distribution of 80% and 20%, respectively. The training set contains 5,515,575 attack samples and 213,090 benign samples, whereas the test set contains 1,378,892 attack samples and 53,274 benign samples. Another significant aspect of this synthetic dataset is the lack of duplicate rows, ensuring the quality of the data and preventing any bias in the model. This distribution is the best benchmark for the efficacy of IDS for various types of IoMT cyber threats.

The synthetic dataset includes the unique distribution of the flags sent during the TCP handshake for various types of attack methodologies, with the values directly obtained from the tshark output for the analysis of the 374 PCAP files from the CICIoMT2024 dataset, ensuring the genuineness of the data.

1. SYN Flood Attacks: DDoS_SYN and DoS_SYN have almost exclusive SYN flags (0.9999), representing the characteristics of pure SYN Flood attacks, where the connection requests are never completed. The flags Acknowledgement (ACK), FIN, PSH, and RST have a fixed value of 0.0, confirming the completion of the handshake is not possible.
2. Null Flag Flooding: Both the DDoS_TCP and the DoS_TCP attacks involve the null flag technique, which is equivalent to a TCP segment with all flags set to 0.0.
3. Reconnaissance: Recon_Port_Scan has a higher SYN flag rate of 0.63, accompanied by a higher RST packet rate of 0.32, indicating a SYN scan. Recon_OS_Scan has a balanced profile with a higher SYN flag rate of 0.50 and a higher ACK flag rate of 0.37. Recon_Vuln_Scan has a higher SYN flag rate of 0.42, a higher ACK flag rate of 0.39, and a higher PSH flag rate of 0.15, indicating a higher number of established connections.
4. MQTT Protocol Attacks: In the MQTT_DDoS_Connect and MQTT_DoS_Connect attacks, the ACK flags are dominant, with a rate of 0.55 and 0.58, respectively, accompanied by a higher SYN flag rate of 0.27 and 0.30. In the MQTT_DDoS_Publish and MQTT_DoS_Publish attacks, the PSH flags are dominant, with a rate of 0.42 and 0.45, respectively, accompanied by a higher ACK flag rate of 0.38 and 0.40. In the MQTT_Malformed attack, the PSH flags are dominant, with a rate of 0.35, accompanied by a higher ACK flag rate of 0.40.

In order to ensure semantic correctness and provide strong features for classification, attacks are limited to PCAP-verified network protocols and quantified using characteristic packet attributes.

1. Protocol Enforcement Using IP Protocol Numbers: Protocol_Type relies upon the use of IP protocol numbers (6 = TCP, 17 = UDP, 1 = ICMP) observed in the PCAP captures. DDoS_ICMP and DoS_ICMP use protocol 1 (ICMP). DDoS_UDP and DoS_UDP use protocol 17 (UDP). TCP-based attacks, including all forms of MQTT, SYN floods, and reconnaissance attacks, use protocol 6 (TCP). ARP_Spoofing uses TCP/UDP protocol in the IP layer but sets the ARP indicator flag to 1.0, which indicates the use of Layer 2 ARP in conjunction with the observed IP layer protocol in the PCAP captures.
2. Packet Size Distributions Derived from PCAP Measurements: Packet sizes are directly obtained from tshark tools using frame length analysis of the captures. DDoS_ICMP produces uniformly small packet sizes with an average of ~60 bytes and a maximum of 98 bytes, which is efficient for ICMP echo floods.

ARP_Spoofing produces an average of 482 bytes with the entire range from 42 to 1514 bytes, which is mixed ARP and data traffic. Benign traffic averages ~340 bytes across the entire MTU range from 42 to 1514 bytes, which indicates diverse IoMT device communications. DDoS_UDP produces average packet sizes of ~353 bytes, while Recon_Ping_Sweep produces average packet sizes of ~179 bytes with maximum sizes of 1046 bytes.

3. Traffic Rate Profiles: Each classification profile has PCAP-derived rate characteristics. DDoS_UDP maintains the highest rate of ~519 packets/s, followed by DDoS_TCP with ~281 packets/s and DDoS_SYN with ~224 packets/s. Reconnaissance attacks maintain moderate rates: Port_Scan maintains ~144 packets/s, while Ping_Sweep maintains ~139 packets/s. Benign traffic maintains a baseline rate of ~69 packets/s, which clearly differentiates from the attacks.

6 Evaluation Results and Discussion

The increasing trend towards the adoption of machine learning (ML) and deep learning (DL) for various cybersecurity-related applications necessitates the evaluation of the performance of ML models under various conditions. For this purpose, several ML algorithms have been evaluated, including the ensemble method Random Forest, AdaBoost, the linear method Logistic Regression, and the deep learning method Deep Neural Network. For the evaluation, four ML algorithms have been implemented, each with their optimal parameters. We selected Random Forest, AdaBoost, Logistic Regression, and Deep Neural Networks to ensure direct comparability with the benchmark results reported on the real-world IoMT dataset CICIoMT2024 in [31]. Since one goal of our study is to evaluate whether a synthetic dataset can reproduce the relative performance rankings and absolute detection metrics of the real dataset, using the same set of algorithms eliminates model selection as a confounding variable. This allows us to isolate the effect of replacing real data with synthetic data.

For instance, the Logistic Regression method, which is the baseline linear method, has been implemented with L2 regularization, C = 1.0, the L-BFGS solver, and a maximum of 100 iterations. AdaBoost, another ensemble method, has been implemented with 50 estimators, the SAMME.R algorithm, and the learning rate set to 1.0. Moreover, the Random Forest method, another ensemble method, has been implemented with 100 estimators, the Gini impurity criterion, the minimum samples split criterion set to 2, and the square root of the number of features considered for each split. Lastly, the Deep Neural Network method, which is the baseline method, has been implemented with three hidden layers, each containing 32 neurons, the Adam optimizer, the learning rate set to 0.001, L2 regularization, alpha = 0.0001, and the Nesterov momentum. Moreover, the random seed for the random number generator for each method is fixed at 42.

A key goal of this work was to evaluate the effectiveness of data balancing techniques, namely ADASYN, Sample Weighting, and a combination of SMOTE and SMOTEEN, on both the original CICIoMT2024 dataset and the artificially created SynIoMT2026 dataset, with balanced and unbalanced cases. Classification models were trained on three classification problems of varying difficulty, including binary classification (benign vs. malicious), 6-class classification (major categories of attacks), and a finer-grained classification problem with 19 classes (specific attack types).

6.1 Performance on the CICIoMT2024 Dataset

The models were first created and tested using the actual-world CICIoMT2024 dataset, with the purpose of establishing a baseline for their performance and determining the need for data balancing in an imbalanced context. The results show how data balancing affects the models' performance in a non-linear way, depending on the complexity of the task performed.

The results for the binary classification problem with ADASYN (Table 4) show how all models perform exceptionally well with the imbalanced data, with all accuracy and F1-score results higher than 99.5%. The Random Forest model maintains its high performance even after balancing the data (Accuracy: 99.86%, F1-score: 99.86%), with a slight improvement in both metrics by +0.007%. The Logistic Regression and Deep Neural Network models show a common pattern in how they perform with and without balancing, where the balanced data results in a slightly lower accuracy (−1.05%, −0.03%, respectively) and high F1-scores (>99.6%), indicating a balanced performance for both classes. The ADASYN method results in a significant increase in training time (2–4 times) for all models, with the longest training time for the Logistic Regression method, lasting 216 min.

Table 4: ADASYN balancing technique on CICIoMT2024 dataset.

Classification	Model	Data	Accuracy	Precision	Recall	F1-Score	Time (min)
Binary	LR	Imbalance	0.9955	0.9955	0.9955	0.9955	162
		Balance	0.9850	0.9908	0.9850	0.9867	216
	AdaBoost	Imbalance	0.9984	0.9985	0.9984	0.9984	8
		Balance	0.9981	0.9982	0.9981	0.9981	28
	RF	Imbalance	**0.9986**	**0.9985**	**0.9986**	**0.9986**	1
		Balance	**0.9986**	**0.9986**	**0.9986**	**0.9986**	4
	DNN	Imbalance	0.9967	0.9966	0.9967	0.9966	22
		Balance	0.9963	0.9964	0.9963	0.9964	40
6-Class	LR	Imbalance	0.9302	0.9366	0.9302	0.9326	6
		Balance	0.8312	0.9305	0.8312	0.8638	11
	AdaBoost	Imbalance	0.9413	0.9414	0.9413	0.9412	1
		Balance	0.7226	0.8441	0.7226	0.7097	1
	RF	Imbalance	**0.9751**	**0.9764**	**0.9751**	**0.9756**	0
		Balance	**0.9644**	**0.9688**	**0.9644**	**0.9658**	0
	DNN	Imbalance	0.9517	0.9514	0.9517	0.9515	3
		Balance	0.9114	0.9461	0.9114	0.9232	9
19-Class	LR	Imbalance	0.7603	0.7098	0.7603	0.6755	370
		Balance	0.7277	0.6808	0.7277	0.6704	1620
	AdaBoost	Imbalance	0.2364	0.4388	0.2364	0.1433	12
		Balance	0.2453	0.5090	0.2453	0.1497	89
	RF	Imbalance	**0.9953**	**0.9958**	**0.9953**	**0.9951**	1
		Balance	**0.9901**	**0.9932**	**0.9901**	**0.9895**	10
	DNN	Imbalance	0.9861	0.9880	0.9861	0.9841	30
		Balance	0.7325	0.7218	0.7325	0.7007	191

Note: Bold numbers indicate the highest value among either all **Imbalance** approaches or all **Balance** approaches separately within the same classification type (Binary, 6-Class, or 19-Class) for each metric and model category.

In the 6-class classification, the results of the data balancing were more significant. Random Forest once again showed significant stability, obtaining high performance (Accuracy: 96.44%, F1: 96.58%) with only slight deterioration after balancing. Logistic Regression and AdaBoost showed significant performance deterioration, with accuracy decreasing by 9.9% and 21.9%, respectively, suggesting sensitivity to the introduction of synthetic samples in the multi-class classification setting. The DNN also showed significant deterioration −4.03% accuracy, −2.82% F1 suggesting problems in learning from balanced data for tasks of intermediate complexity.

In the 19-class classification, Random Forest again showed outstanding performance (Accuracy: 99.01%, F1: 98.95%) despite the increased complexity of the task, with only slight deterioration (−0.52% accuracy). The Deep Neural Network showed a dramatic performance deterioration, with accuracy decreasing from 98.61% to 73.25% (−25.36%) and the F1-score decreasing by 28.34%, suggesting fundamental incompatibility with ADASYN balancing in high-complexity classification tasks. Although the absolute performance is low, AdaBoost shows relative improvement in performance with balancing +0.89% accuracy, +0.64% F1.

The efficacy of the ADASYN balancing technique is shown to decline with an increase in the complexity of the task for the majority of the models, with adverse effects being particularly pronounced for the application of the technique in the context of the Deep Learning approach for fine-grained classification. Overall, the results suggest that the efficacy of the ADASYN balancing technique is heavily dependent on the model type and the complexity of the classification task, with the Random Forest approach being the most reliable for balanced and imbalanced datasets.

In Table 5, the Sample Weighting approach was shown to have high computational efficiency, with the approach having negligible effects on high-performing models for the binary classification problem. For the AdaBoost approach, the performance metrics remained the same with or without the application of the sample weighting approach, indicating near-perfect stability. Similar results were obtained for the Deep Neural Network approach, where the accuracy remained at approximately 99.67% with the application of the sample weighting approach. For the Random Forest approach, the accuracy remained the same with a minor reduction of 0.02% in the performance metrics. For the Logistic Regression approach, the accuracy was reduced significantly, but the F1-score remained high. For the 6-class classification problem, the accuracy remained the same for the Random Forest approach, whereas the Logistic Regression approach had a significant reduction in accuracy. However, the precision for the Logistic Regression approach was significantly enhanced. For the AdaBoost approach, the results remained the same, indicating the approach's compatibility with the original class distribution. For the 19-class classification problem, the Logistic Regression approach had the accuracy reduced significantly, indicating the limitations of the approach. However, the Random Forest approach had an excellent accuracy of 99.24%, indicating high resilience. For the AdaBoost approach, the results remained the same, indicating the approach's compatibility with the original class distribution.

Table 5: Sample weighting balancing technique on CICIoMT2024 dataset.

Classification	Model	Data	Accuracy	Precision	Recall	F1-Score	Time (min)
Binary	LR	Imbalance	0.9955	0.9955	0.9955	0.9955	193
		Balance	0.9857	0.9911	0.9857	0.9873	98
	AdaBoost	Imbalance	0.9984	0.9985	0.9984	0.9984	8
		Balance	**0.9984**	**0.9985**	**0.9984**	**0.9984**	8

(Continued)

Table 5 (continued)

Classification	Model	Data	Accuracy	Precision	Recall	F1-Score	Time (min)
	RF	Imbalance	**0.9986**	**0.9985**	**0.9986**	**0.9986**	1
		Balance	0.9984	0.9983	0.9984	0.9983	1
	DNN	Imbalance	0.9967	0.9966	0.9967	0.9966	21
		Balance	0.9967	0.9966	0.9967	0.9966	21
6-Class	LR	Imbalance	0.9302	0.9366	0.9302	0.9326	6
		Balance	0.8724	0.9570	0.8724	0.9063	5
	AdaBoost	Imbalance	0.9413	0.9414	0.9413	0.9412	1
		Balance	0.9413	0.9414	0.9413	0.9412	1
	RF	Imbalance	**0.9751**	**0.9764**	**0.9751**	**0.9756**	0
		Balance	**0.9753**	**0.9755**	**0.9753**	**0.9754**	0
	DNN	Imbalance	0.9517	0.9514	0.9517	0.9515	3
		Balance	0.9517	0.9514	0.9517	0.9515	3
19-Class	LR	Imbalance	0.7603	0.7098	0.7603	0.6755	481
		Balance	0.5953	0.7076	0.5953	0.5603	489
	AdaBoost	Imbalance	0.2364	0.4388	0.2364	0.1433	13
		Balance	0.2364	0.4388	0.2364	0.1433	14
	RF	Imbalance	**0.9953**	**0.9958**	**0.9953**	**0.9951**	1
		Balance	**0.9924**	**0.9944**	**0.9924**	**0.9912**	1
	DNN	Imbalance	0.9861	0.9880	0.9861	0.9841	33
		Balance	0.9861	0.9880	0.9861	0.9841	33

Note: Bold numbers indicate the highest value among either all **Imbalance** approaches or all **Balance** approaches separately within the same classification type (Binary, 6-Class, or 19-Class) for each metric.

Sample Weighting was observed to always provide training time that was equal to or less than the training time obtained using other techniques for all scenarios, in contrast to ADASYN, which significantly increases the computational cost. Sample Weighting maintains the performance of the majority of the models for binary and 6-class classification problems, in contrast to ADASYN, which significantly degrades the performance of the models for complex classification problems like 19-class classification, especially for Deep Neural Networks (DNNs). The performance of the Random Forest model was observed to be significantly high for both techniques, with ADASYN showing slightly better performance for binary classification problems and Sample Weighting showing better computational efficiency. The Deep Neural Network was completely unaffected by Sample Weighting, while the performance of the model was catastrophic for the 19-class classification scenario using ADASYN. The performance of the Logistic Regression model was observed to be facing difficulties with the application of both techniques for complex classification problems like

19-class classification, showing severe degradation for the 19-class classification scenario. For binary classification problems, the techniques are equally applicable, with Sample Weighting showing computational advantages. In the context of complex classification problems, Sample Weighting is recommended for Deep Learning techniques, while the Random Forest model shows robust performance for both techniques. In the context of computational efficiency in resource-constrained environments, Sample Weighting shows better computational efficiency.

The usage of the hybrid method, as shown in Table 6, indicates that the method's performance characteristics show promising results for the model architectures, along with high computational costs. For the binary classification problem, the results show that all models have better precision, with AdaBoost achieving near-perfect precision at 99.999%, while Logistic Regression achieved an increase of +0.21%. However, the accuracy increase for the Random Forest and DNN models was less significant at +0.008% and +0.015%, respectively. Most models also had a slight decline in recall, particularly for the Logistic Regression model, which had the largest decline at −1.19%. Moreover, the training time for the models significantly increased, particularly for the Logistic Regression model, which had an increase of 79% at 209 min, while the training time for the DNN classifer doubled to 41 min.

Table 6: SMOTE+SMOTEEN balancing technique on CICIoMT2024 dataset.

Classification	Model	Data	Accuracy	Precision	Recall	F1-Score	Time (min)
Binary	LR	Imbalance	0.9955	0.9978	0.9976	0.9977	117
		Balance	0.9860	0.9999	0.9858	0.9928	210
	AdaBoost	Imbalance	0.9984	**0.9993**	0.9991	0.9992	8
		Balance	0.9982	**1.0000**	0.9982	0.9991	26
	RF	Imbalance	**0.9986**	0.9991	**0.9995**	**0.9993**	1
		Balance	**0.9987**	0.9995	**0.9992**	**0.9993**	3
	DNN	Imbalance	0.9966	0.9978	0.9987	0.9982	19
		Balance	0.9967	0.9990	0.9976	0.9983	41
6-Class	LR	Imbalance	0.7693	0.7844	0.7693	0.7090	169
		Balance	0.6639	0.6751	0.6639	0.6688	708
	AdaBoost	Imbalance	0.6979	0.8547	0.6979	0.7094	8
		Balance	0.9165	0.9271	0.9165	0.9127	65
	RF	Imbalance	**0.9988**	**0.9989**	**0.9988**	**0.9988**	1
		Balance	**0.9983**	**0.9984**	**0.9983**	**0.9984**	8
	DNN	Imbalance	0.9946	0.9959	0.9946	0.9951	37
		Balance	0.9903	0.9928	0.9903	0.9913	129
	LR	Imbalance	0.7603	0.7098	0.7603	0.6755	366
		Balance	0.5933	0.7060	0.5933	0.5576	1629

(Continued)

Table 6 (continued)

Classification	Model	Data	Accuracy	Precision	Recall	F1-Score	Time (min)
19-Class	AdaBoost	Imbalance	0.2364	0.4388	0.2364	0.1433	12
		Balance	0.5194	0.6466	0.5194	0.4364	87
	RF	Imbalance	**0.9953**	**0.9958**	**0.9953**	**0.9951**	1
		Balance	**0.9919**	**0.9938**	**0.9919**	**0.9909**	9
	DNN	Imbalance	0.9861	0.9880	0.9861	0.9841	30
		Balance	0.9673	0.9715	0.9673	0.9670	173

Note: Bold numbers indicate the highest value among either all **Imbalance** approaches or all **Balance** approaches separately within the same classification type (Binary, 6-Class, or 19-Class) for each metric.

For the 6-class classification problem, the results show that the method achieved a significant increase in accuracy for the AdaBoost model, increasing accuracy from 69.79% to 91.65% (+21.86%), while the F1-score also increased from 70.94% to 91.27% (+20.33%). For the Random Forest model, the accuracy remained high at 99.83%, with a slight decline of −0.046%. However, the Logistic Regression model had a significant decline in performance, particularly at −10.54% accuracy, as well as −4.02% for the F1-score. For the 19-class classification problem, the results show that the method achieved a remarkable increase in accuracy for the AdaBoost model, increasing accuracy from 23.64% to 51.94% (+28.29%), while the F1-score also increased from 14.33% to 43.64% (+29.31%). For the Random Forest model, the accuracy remained high at 99.19%, with a slight decline of −0.34%. For the DNN classifier, the results show that the method had an acceptable preservation of performance, with an accuracy of 96.73%.

We can see that Random Forest has a high performance level across all techniques used. The combination of using SMOTE+SMOTEEN has a marginal improvement in precision but at a cost of using more computational resources. There is a significant improvement in using the AdaBoost technique when combined with the combination of using SMOTE+SMOTEEN, while other techniques show a small or zero improvement. The Logistic Regression technique shows a constant level of deterioration when any balancing technique is used in complex problems. In deep learning techniques, the combination of using SMOTE+SMOTEEN shows a stable performance, while the use of ADASYN shows a catastrophic level of failure. The use of the combination of the SMOTE+SMOTEEN technique shows promise as a tool, especially when used to improve the performance of some ensemble techniques in complex multi-class problems, but at a cost of using more computational resources.

Computational Time Analysis of Balancing Techniques on the CICIoMT2024 Dataset

With respect to computational overhead, the implementation of ADASYN for balancing leads to higher computational time for training for each of the models and classifiers, although the degree of increase in training time depends on the complexity of a model and the number of classes. Binary classifiers experience an increase in training time by 1.33 times in logistic regression, 3.5 times in Adaboost, 4 times in random forest, and 1.8 times in DNN. 6-class classification led to an increase in training time by only 0.83 times in logistic regression, and 3 times in DNN while Adaboost and random forest models showed no increase, staying at 1 and 0 min, correspondingly. The highest increase in training time was observed for the 19-class classifiers – logistic regression trained by ADASYN required 4.4 times more time than without balancing, 7.4 times for Adaboost, 10 times for random forest, and 6.4 times for DNN. This results suggests that ADASYN

improves class balance and F1-score for each classifier but requires a considerable amount of computation time, especially in high dimensional spaces for logistic regression (1620 min) and Adaboost (7.4 times increase). Thus, both options should be applied cautiously due to high overheads when using ADASYN for balancing. The best choice is, again, the random forest which only increased its training time by 4–10 min.

Contrary to the ADASYN balancing technique, the Sample Weighting (SW) approach demonstrated relatively low computational burden for all models and tasks, with some models being characterized by equal training time in the imbalanced and balanced cases. For binary classification, SW decreased training time of Logistic Regression from 193 to 98 min (decrease of about 49.2%), while AdaBoost showed equal training time of 8 min for both classes. Random Forest and DNN training times also did not change (1 and 21 min, respectively). In the case of 6-class classification, Logistic Regression training time dropped from 6 to 5 min, whereas AdaBoost, Random Forest, and DNN demonstrated no change in time (1, 0, and 3 min, respectively). Training times for the difficult 19-class classification for Logistic Regression grew slightly from 481 to 489 min (+1.7%), while for AdaBoost, training time rose from 13 to 14 min. At the same time, Random Forest still required 1 min for training, while DNN training time increased from 32 to 33 min. The data demonstrates clearly that SW training was more computationally economical than ADASYN. Furthermore, it shows that the former approach did not add any or added insignificant training time for the model while providing comparable or even better classification quality (e.g., Random Forest achieved 99.24% accuracy for 19 classes but required 1 min for training). As such, SW can be used for IoT-based IDS design.

However, SMOTE+SMOTEEN balancing method showed high overheads in terms of computations among almost all classifiers under consideration, especially in more complex tasks. In binary classification, the training time for Logistic Regression increased from 117 to 210 min (1.8×), AdaBoost from 8 to 26 min (3.25×), Random Forest from 1 to 3 min (3×), and DNN from 19 to 41 min (2.16×). In 6-classification tasks, the overhead became even higher: Logistic Regression took from 169 to 708 min (4.2×), AdaBoost from 8 to 65 min (8.1×), Random Forest from 1 to 8 min (8×), and DNN from 37 to 129 min (3.5×). The highest overhead was observed in 19-class classification tasks: Logistic Regression from 366 to 1629 min (4.45×), AdaBoost from 12 to 87 min (7.25×), Random Forest from 1 to 9 min (9×), and DNN from 30 to 173 min (5.8×). At the same time, SMOTE+SMOTEEN showed both positive and negative effects on model performance. Specifically, in 19-class classification, AdaBoost demonstrated improved performance with an increase in accuracy from 23.6% to 51.9%, but Logistic Regression and DNN showed performance degradation (e.g., DNN accuracy dropped from 98.6% to 96.7%). Overall, one can conclude that although SMOTE+SMOTEEN proved to be useful for specific classifiers, e.g., AdaBoost with multi-class classifications, its significant overhead in terms of computation time makes its use impractical for some models, especially Logistic Regression with more than 27 h (1629 min) required for one training run.

6.2 Performance on the SynIoMT2026 Dataset

This section is dedicated to providing an evaluation of balancing techniques on the synthetic dataset SynIoMT2026, which provides vital insights into how synthetic data handle class imbalance mitigation techniques compared to the original dataset, CICIoMT2024.

The application of ADASYN data balancing on the SynIoMT2026 dataset, as illustrated in Table 7, showed varied performance levels across classification tasks and models. In the binary classification task, all models, including LR, AdaBoost, RF, and DNN, showed perfect performance, denoted by a score of 1.0, on all performance metrics for imbalanced and balanced data. This indicates that classification is trivial for this dataset, regardless of balancing. However, as classification complexities increase, ADASYN balancing showed improved performance. In the 6-class classification, Random Forest showed superior performance, achieving near-perfect performance on imbalanced data with minimal training time, denoted by 2 min, and

showed reduced performance after balancing. In contrast, other models, including LR, showed impressive performance on imbalanced data but had substantial training times, denoted by 119 min, which significantly increased after balancing, denoted by 479 min, with slight performance degradation. In contrast, AdaBoost showed poor performance, denoted by F1-scores < 0.47, regardless of balancing, indicating that it is not applicable to this dataset's complexity level. In the 19-class classification, which is more complex, Random Forest showed impressive performance, achieving solid performance on imbalanced data with minimal training time, denoted by 4 min, and moderate performance degradation after balancing. In contrast, LR and DNN showed impressive performance on imbalanced data, but performance significantly decreased after balancing, with substantial increases in training times, especially LR, which exceeded 1500 min. In conclusion, ADASYN balancing showed improved performance by addressing class imbalance, but it decreased classification performance and substantially increased training times for all models, except Random Forest, which showed impressive performance and minimal training times.

Table 7: ADASYN balancing technique on SynIoMT2026 dataset.

Classification	Model	Data	Accuracy	Precision	Recall	F1-Score	Time (min)
Binary	LR	Imbalance	1	1	1	1	5
		Balance	1	1	1	1	5
	AdaBoost	Imbalance	1	1	1	1	0
		Balance	1	1	1	1	0
	RF	Imbalance	1	1	1	1	1
		Balance	1	1	1	1	1
	DNN	Imbalance	1	1	1	1	16
		Balance	1	1	1	1	16
6-Class	LR	Imbalance	0.9963	0.9963	0.9963	0.9963	118
		Balance	**0.9710**	**0.9722**	**0.9710**	**0.9708**	479
	AdaBoost	Imbalance	0.5690	0.4254	0.5690	0.4613	39
		Balance	0.5089	0.4109	0.5089	0.4145	105
	RF	Imbalance	**0.9992**	**0.9992**	**0.9992**	**0.9992**	2
		Balance	0.8034	0.8619	0.8034	0.8135	6
	DNN	Imbalance	0.9981	0.9981	0.9981	0.9981	80
		Balance	0.9136	0.9159	0.9136	0.9099	68
19-Class	LR	Imbalance	0.8879	0.8880	0.8879	0.8879	292
		Balance	**0.7673**	0.7664	**0.7673**	**0.7266**	1502
	AdaBoost	Imbalance	0.5723	0.4666	0.5723	0.4628	78
		Balance	0.3761	0.4103	0.3761	0.2894	192

(Continued)

Table 7 (continued)

Classification	Model	Data	Accuracy	Precision	Recall	F1-Score	Time (min)
	RF	Imbalance	**0.8916**	**0.8916**	**0.8916**	**0.8916**	4
		Balance	0.7314	**0.8243**	0.7314	0.6985	13
	DNN	Imbalance	0.8901	0.8542	0.8901	0.8632	149
		Balance	0.6955	0.6238	0.6955	0.6296	147

Note: Bold numbers indicate the highest value among either all **Imbalance** approaches or all **Balance** approaches separately within the same classification type (Binary, 6-Class, or 19-Class) for each metric.

The application of the Sample Weighting balancing technique to the SynIoMT2026 dataset, as shown in Table 8, indicates diverse performance characteristics for the classification problem and the models applied. For the binary classification problem, all the models, namely Logistic Regression, AdaBoost, Random Forest, and Deep Neural Networks, performed with a perfect score of 1.0 for all metrics when the data is both imbalanced and balanced, indicating that the problem of binary threat detection using this synthetic dataset is solved irrespective of the balancing approach applied. The training time for all the models is also very low, with AdaBoost having the lowest time of less than a minute, and the DNN having the highest time of 14 min. For the complex problem of 6-class classification, the Random Forest approach performed extremely well with a near-perfect score of 0.9992 for both imbalanced and balanced data, with remarkably low training times of only 2 min, making it the best approach for solving this problem. The Logistic Regression approach also performed well with a score of 0.9963 for all the metrics for both imbalanced and balanced data, although with extremely high training costs of 119 and 126 min, respectively. The Deep Neural Networks approach performed well with a score of 0.9981 for all the metrics irrespective of the balancing approach applied, although with moderate training costs of 79 min. The AdaBoost approach, however, performed poorly for the 6-class problem, with its F1-score reducing from 0.4613 for imbalanced data to as low as 0.1663 after sample weighting, indicating its absolute inability to solve this problem.

Table 8: Sample weighting balancing technique on SynIoMT2026 dataset.

Classification	Model	Data	Accuracy	Precision	Recall	F1-Score	Time (min)
Binary	LR	Imbalance	1	1	1	1	7
		Balance	1	1	1	1	6
	AdaBoost	Imbalance	1	1	1	1	0
		Balance	1	1	1	1	0
	RF	Imbalance	1	1	1	1	1
		Balance	1	1	1	1	1
	DNN	Imbalance	1	1	1	1	14
		Balance	1	1	1	1	14
	LR	Imbalance	0.9963	0.9963	0.9963	0.9963	119
		Balance	0.9962	0.9962	0.9962	0.9962	126

(Continued)

Table 8 (continued)

Classification	Model	Data	Accuracy	Precision	Recall	F1-Score	Time (min)
6-Class	AdaBoost	Imbalance	0.5690	0.4254	0.5690	0.4613	40
		Balance	0.2997	0.1349	0.2997	0.1663	38
	RF	Imbalance	**0.9992**	**0.9992**	**0.9992**	**0.9992**	2
		Balance	**0.9992**	**0.9992**	**0.9992**	**0.9992**	2
	DNN	Imbalance	0.9981	0.9981	0.9981	0.9981	79
		Balance	0.9981	0.9981	0.9981	0.9981	79
19-Class	LR	Imbalance	0.8879	0.8880	0.8879	0.8879	286
		Balance	0.8879	**0.8880**	0.8879	**0.8879**	286
	AdaBoost	Imbalance	0.5723	0.4666	0.5723	0.4628	73
		Balance	0.4645	0.3637	0.4645	0.3886	74
	RF	Imbalance	**0.8915**	**0.8915**	**0.8915**	**0.8915**	4
		Balance	0.8412	0.8656	0.8412	0.8270	4
	DNN	Imbalance	0.8901	0.8542	0.8901	0.8632	145
		Balance	**0.8901**	0.8542	**0.8901**	0.8632	145

Note: Bold numbers indicate the highest value among either all **Imbalance** approaches or all **Balance** approaches separately within the same classification type (Binary, 6-Class, or 19-Class) for each metric.

In Table 8, in the challenging 19-class classification problem, the performance of the Random Forest algorithm was once more outstanding, with an F1-score of 0.8915 even in the presence of imbalanced data with minimum training time (4 min). The use of balanced data resulted in a slightly decreased performance to the range of 0.8270–0.8656. In the case of Logistic Regression, the performance remained almost stable with an F1-score of 0.8879 in both cases, with high training times of 286 min. The performance of the Deep Neural Networks was comparable to the performance of the Random Forest algorithm in the presence of imbalanced data with an F1-score of 0.8632. In this case, the Sample Weighting technique did not affect the performance of the algorithm; in fact, the performance with the use of balanced data was even better with the same F1-score of 0.8632 and the same training time of 145 min. Sample Weighting was a highly efficient technique in balancing the dataset with the least training time while even slightly improving the performance in some cases, with the Random Forest algorithm being the most efficient algorithm in the entire range of classification complexities.

The hybrid SMOTE+SMOTEEN method showed promising performance characteristics for all the classification tasks involving the SynIoMT2026 dataset, as depicted in Table 9. For the binary classification problem, all the models involving Logistic Regression, AdaBoost, Random Forest, and Deep Neural Networks achieved a perfect score of 1.0 for all the evaluation parameters for both imbalanced and balanced data, indicating the simplicity of the binary threat detection problem regardless of the balancing technique applied. Moreover, SMOTE+SMOTEEN significantly reduced the training time for balanced data for all the models, with Logistic Regression reducing training time from 6 to 1 min and Deep Neural Networks reducing training time from 17 to 2 min, indicating the efficiency of the technique in terms of computational

time. For the multi-class problem, Random Forest once again showed promising results, achieving near-perfect scores for imbalanced data (0.9992) and extremely high scores for balanced data (0.9975) with the added advantage of completing the problem in under 3 min, with balanced data training time being zero. Logistic Regression also showed robust results with slightly reduced performance after balancing, reducing from 0.9962 to 0.9867 for all parameters, with a significant reduction in training time from 110 to 14 min. Similarly, Deep Neural Networks showed reduced balanced data performance, reducing slightly from 0.9975 to 0.9891, with a significant reduction in training time from 64 to 6 min. Interestingly, AdaBoost showed consistent results for all parameters, with the F1-score ranging from 0.71 to 0.78 for imbalanced and balanced data, with reduced balanced data training time from 57 to 6 min, indicating the technique's ability to improve AdaBoost's computational efficiency despite being unable to improve its inherent limitations for multi-class problem-solving. For the more challenging 19-class problem, Random Forest again showed promising results, with balanced data performance being robust with an F1-score of 0.8862 compared to imbalanced data, with the added advantage of completing balanced data training in zero measurable time. Logistic Regression showed moderate performance degradation after data balancing, as its F1-score reduced from 0.8879 to 0.8597. However, it showed considerable reduction in training time, from 305 to 19 min. Similarly, Deep Neural Networks showed similar trends; data balancing resulted in an F1-score of 0.8634, which is almost identical to 0.8632, as seen with imbalanced data. At the same time, training time reduced significantly from 150 to 5 min. In all tasks, SMOTE+SMOTEEN showed its major strength: reduced training cost with minimal performance degradation. In this synthetic dataset, Random Forest showed its strength as the best-performing model for all classification complexities.

Table 9: SMOTE+SMOTEEN balancing technique on SynIoMT2026 dataset.

Classification	Model	Data	Accuracy	Precision	Recall	F1-Score	Time (min)
Binary	LR	Imbalance	1	1	1	1	6
		Balance	1	1	1	1	1
	AdaBoost	Imbalance	1	1	1	1	0
		Balance	1	1	1	1	0
	RF	Imbalance	1	1	1	1	1
		Balance	1	1	1	1	0
	DNN	Imbalance	1	1	1	1	17
		Balance	1	1	1	1	2
6-Class	LR	Imbalance	0.9962	0.9962	0.9962	0.9962	110
		Balance	0.9867	0.9867	0.9867	0.9867	14
	AdaBoost	Imbalance	0.7845	0.6768	0.7845	0.7127	57
		Balance	0.7845	0.6768	0.7845	0.7127	6
	RF	Imbalance	**0.9992**	**0.9992**	**0.9992**	**0.9992**	3
		Balance	**0.9975**	**0.9975**	**0.9975**	**0.9975**	0
	DNN	Imbalance	0.9975	0.9975	0.9975	0.9975	64
		Balance	0.9891	0.9891	0.9891	0.9891	6

(Continued)

Table 9 (continued)

Classification	Model	Data	Accuracy	Precision	Recall	F1-Score	Time (min)
19-Class	LR	Imbalance	0.8879	0.8880	0.8879	0.8879	305
		Balance	0.8597	0.8622	0.8597	0.8597	19
	AdaBoost	Imbalance	0.5723	0.4666	0.5723	0.4628	83
		Balance	0.5151	0.3553	0.5151	0.5151	6
	RF	Imbalance	**0.8916**	**0.8916**	**0.8916**	**0.8916**	4
		Balance	**0.8862**	**0.8865**	**0.8862**	**0.8862**	0
	DNN	Imbalance	0.8901	0.8542	0.8901	0.8632	150
		Balance	0.8634	0.8656	0.8634	0.8634	5

Note: Bold numbers indicate the highest value among either all **Imbalance** approaches or all **Balance** approaches separately within the same classification type (Binary, 6-Class, or 19-Class) for each metric.

A comparative analysis of the three balancing techniques, namely, ADASYN, Sample Weighting, and SMOTE+SMOTEEN, on the SynIoMT2026 dataset showed that each of the machine learning algorithms had its own characteristics in terms of performance. For the 2-class problem, all the models had perfect performance, i.e., 1.0, for all the metrics irrespective of the balancing technique used, which indicates that the problem of binary threat detection is well-handled. For the 6- and 19-class problems, the Random Forest model was the best-performing model for all the balancing techniques. It achieved its best performance for the 6-class problem using the ADASYN technique, followed closely by the 19-class problem. For the 19-class problem, the Random Forest model achieved its best performance using the Sample Weighting technique, followed closely by the SMOTE+SMOTEEN technique. However, the Random Forest model was the most robust, as its performance was near optimal for all the balancing techniques. For the 19-class problem, the performance of the Random Forest model was better for the Sample Weighting technique, whereas the SMOTE+SMOTEEN technique was better for the 19-class problem in terms of the computational cost, which was zero min for the balanced 19-class problem. For the Logistic Regression model, the best performance was achieved for the 6-class problem using the ADASYN technique, followed closely by the 19-class problem. However, the SMOTE+SMOTEEN technique was the best for the Logistic Regression model in terms of the computational cost, as the 6-class balanced problem training time was reduced from 479 min for the ADASYN technique and 126 min for the Sample Weighting technique to 14 min, whereas the 19-class balanced problem training time was reduced from 1502 min for the ADASYN technique and 286 min for the Sample Weighting technique to 19 min—a reduction of 98%. For the 19-class problem, the Logistic Regression model had the least performance degradation. DNN's best 6-class results were achieved using Sample Weighting on the imbalanced dataset, with an F1-score of 0.9981. For the 19-class results on the imbalanced dataset, the best results were achieved using ADASYN, which had an F1-score of 0.8632. In terms of speed, the SMOTE+SMOTEEN approach was the most efficient for the DNN. For the 6-class balanced dataset, the training time was reduced from 80 min using the ADASYN approach, or 79 min using the Sample Weighting approach, to just 6 min. For the 19-class balanced dataset, the training time was reduced from 147 min, 148 min, or 149 min to 5 min, representing a 96% reduction, with the F1-scores being almost. AdaBoost was behind in all cases when it came to solving multi-class problems. Its best performance was on the 6-class F1 score, which was 0.7127, using the combination of SMOTE+SMOTEEN

on both imbalanced and balanced datasets. On the 19-class dataset, its best performance was an F1 score of 0.4628, using ADASYN on the imbalanced dataset. Of all the techniques, Sample Weighting had the lowest performance for AdaBoost, with the balanced dataset on the 6-class dataset having a poor F1 score of 0.1663. Overall, the combination of SMOTE+SMOTEEN was seen to be the most computationally efficient, reducing training times significantly while still maintaining good performance. The technique of Sample Weighting was seen to have the best transfer of performance to Random Forest and Logistic Regression on more complex datasets, while ADASYN had the best performance peaks but at a much higher computational cost.

A comparative evaluation of balancing techniques was carried out using two datasets, namely the real-world CICIoMT2024 benchmark dataset and the synthetic SynIoMT2026 dataset. The results obtained from this study provide a compelling narrative. When using the CICIoMT2024 benchmark dataset, Random Forest was seen to perform better than all other models, regardless of the balancing technique employed. When using the SynIoMT2026 dataset, all balancing techniques provided perfect binary classification results, with Random Forest outperforming all other techniques in all cases.

From the study, the following trends were noted across all balancing techniques when using both datasets: (1) Random Forest was seen to be the best model, regardless of the balancing technique employed. (2) The combination of SMOTE+SMOTEEN was seen to perform better than all other techniques, especially in boosting the performance of weaker models, such as AdaBoost. (3) Sample Weighting was seen to maintain the performance of top-performing models after balancing. (4) ADASYN was seen to provide balanced but mediocre results across all cases. (5) The SynIoMT2026 dataset was seen to provide a perfect replica of the real-world dataset trends, while at the same time being significantly easier to work with, especially when it comes to balancing techniques, since it is a synthetic dataset.

This study has shown that SynIoMT2026 can be used in place of CICIoMT2024 when developing machine learning models, while at the same time providing a compelling narrative on how balancing techniques can be used, especially when developing machine learning models.

We have also noticed that the usage of real-world data is beneficial in boosting the accuracy of bump detection, which again emphasizes the need to handle class imbalance problems in the field of IoMT intrusion detection systems. Nevertheless, there is a major trade-off in the sense that the larger the disparity between classes in real-world data, such as CICIoMT2024, the greater the computational challenge involved. For instance, boosting the minority classes through techniques like ADASYN results in increased time and memory consumption during the training phase due to the increased number of synthetic instances generated during the balancing process. For instance, during the experiments conducted using the CICIoMT2024 dataset, the balancing through ADASYN resulted in increased time consumption during the training phase of the Logistic Regression model for the 19-class classification task, reaching up to 1620 min, whereas the original time consumption was only a fraction of this value. This again emphasizes the need to rely on synthetic datasets, such as SynIoMT2026, in the sense that the disparity between classes is naturally low, resulting in the balancing process being handled efficiently through any balancing technique with low computational cost.

6.2.1 Computational Time Analysis of Balancing on SynIoMT2026 Dataset

The application of the proposed ADASYN balancing technique on the dataset under consideration produced significantly greater overhead compared to previous results and was characterized by considerable variations in terms of computational costs depending on the task difficulty and type of classifier used. Thus, in binary classification tasks, no significant changes in time were observed: Logistic Regression required 5 min both for the imbalanced and the balanced cases; AdaBoost took no extra time, Random Forest—1 min, and

DNN—16 min, implying that ADASYN has minimal time overhead in binary problems for the considered dataset. On the contrary, for 6-class classification, some time expansions occurred, with Logistic Regression taking 479 min instead of 118 min (4.1× increase), AdaBoost requiring 105 min rather than 39 min (2.7×), and Random Forest needing 6 min instead of 2 min (3×), while the DNN reduced its training time from 80 to 68 min (–15%). Significant expansions of time were registered during the 19-class classification, namely, Logistic Regression took 1502 min in the balanced case against 292 min in the unbalanced one (5.1×), i.e., over 25 h of processing. AdaBoost saw an expansion of time from 78 to 192 min (2.5×), Random Forest—increased from 4 to 13 min (3.25×), while the DNN experienced time reduction from 149 to 147 min. At the same time, although being computationally costly, ADASYN balancing negatively impacted models' performance in terms of accuracy on most multi-classification problems, e.g., the accuracy of Random Forest decreased from 99.9% to 80.3% for 6 classes and from 89.2% to 73.1% for 19 classes. Overall, Random Forest appeared the most efficient in terms of computation costs, consuming just 1–13 min in all tasks, whereas Logistic Regression was extremely costly when balancing 19 classes (1502 min).

Sample weighting (SW) balancing technique applied to SynIoMT2026 dataset exhibited high computational efficiency, posing very limited or no extra computational costs during model training on almost all models and tasks. Namely, in case of binary classification, there was no additional time consumed in training: Logistic Regression was trained in 7 min (imbalanced data) and 6 min (balanced data); AdaBoost required 0 min for training on both datasets; Random Forest required 1 min for training on both datasets; DNN required 14 min on both datasets, meaning that balancing for a binary classification problem is computationally inexpensive. On the other hand, for the 6-class classification, Logistic Regression training time went up from 119 to 126 min (+5.9%), whereas AdaBoost decreased slightly, from 40 to 38 min. However, Random Forest and DNN did not undergo any changes and were trained in 2 and 79 min, respectively, still showing excellent results—namely, Random Forest with the accuracy of 99.92%. In the most challenging case, which is the 19-class classification, Logistic Regression training took 286 min both for the imbalanced and balanced datasets; similarly, the DNN and Random Forest training was conducted in 145 min and 4 min, respectively, while AdaBoost training increased negligibly, from 73 to 74 min. Crucially, compared to ADASYN—which negatively impacted SynIoMT2026 by reducing model performances significantly (for example, Random Forest accuracy dropped from 99.9% to 80.3% in case of 6-class classification)—SW balancing technique did not decrease model performances but either preserved (as in case of Random Forest in 6-class classification, 99.92% accuracy) or even improved them compared to baseline (Random Forest: 84.12%, DNN: 89.01%).

The SMOTE+SMOTEEN balancing technique implemented on the SynIoMT2026 dataset showed a distinct pattern characterized by considerable drops in training times, particularly for multi-class problems—a rather surprising outcome. Indeed, regarding binary classification problems, the time spent on training was reduced or did not increase for all models. Thus, Logistic Regression training time fell from 6 to 1 min (83%), Random Forest remained at zero training time, DNN training time declined from 17 to 2 min (88%), while AdaBoost remained at 0 min. In case of six classes classification problems, the training time decreases significantly: for Logistic Regression, the time fell from 110 to 14 min (87.3%), for AdaBoost, the time declined from 57 to 6 min (89.5%), for Random Forest, it remains zero, while for DNN, it fell from 64 to 6 min (90.6%). Moreover, for the hardest 19-class classification problem, the training time was considerably shortened: thus, the training time for Logistic Regression was reduced from 305 to 19 min (93.8%), for AdaBoost from 83 to 6 min (92.8%), for Random Forest, it remains at zero training time, while for DNN, the training time declined from 150 to 5 min (96.7%). At the same time, the performance degradation was negligible: 99.75% (as opposed to 99.92%) for Random Forest in 6-class classification, and 88.62% (instead of 89.16%) in 19-class classification; as well as 98.91% (DNN for 6-class) and 86.34% (DNN for 19-class). Therefore, the unique feature of SMOTE+SMOTEEN technique consists in a rather noticeable drop in training time

while sustaining high performance. On the contrary, the influence of SMOTE+SMOTEEN technique on CICIoMT2024 data is opposite: indeed, there is a dramatic increase in training time. It might be explained by lower dimensionality, less samples for synthetic generation, etc.

The intrusion detection system designed in this study is considered with deployment in mind and can be used in any IoMT environment, be it a clinic, a hospital, or any connected medical facility. One of the primary goals of this research is to provide a solution for situations when authentic IoMT network traffic data cannot be easily acquired. This is especially true for medical equipment such as infusion pumps, pacemakers, and other patient monitors. As a result, one of the most valuable benefits of our work lies in the possibility of using our synthetic dataset SynIoMT2026 for training intrusion detection models since it would allow making intelligent machines familiar with IoMT specific attacks in absence of sensitive and hard-to-obtain data.

Additionally, our detailed comparison between different dataset processing scenarios (balanced and imbalanced datasets processed by ADASYN, Sample Weighting, and SMOTE+SMOTEEN) will help healthcare professionals use machine learning models for intrusion detection in practice more efficiently. Results of experiments have demonstrated that both the real CICIoMT2024 dataset and the synthetic SynIoMT2026 dataset allow achieving acceptable results when training ML models. To illustrate, the Random Forest classifier trained using the SynIoMT2026 dataset showed 99.92% and 89.16% accuracy when performing six-class and 19-class classifications, respectively, while taking only four min to train. Based on our findings, both types of data may prove useful when implementing intrusion detection systems. Our future research will focus on deploying edge-based intrusion detection systems for hospital network traffic.

7 Conclusion

In conclusion, this study has shown that the selection of an optimal data balancing technique for the purpose of intrusion detection using IoMT is not necessarily a matter of selecting a particular data balancing technique that is better than others, but rather a matter of strategy, depending on the particular circumstances of the application scenario. This has been demonstrated through our comprehensive evaluation of the techniques using both the real-world CICIoMT2024 dataset and the synthetic SynIoMT2026 dataset. This study has shown that the selection of a data balancing technique has a significant impact on not only the performance of the IDS but also on the computational efficiency of the system. The greater the class disparity in the real-world dataset, the greater the computational challenges, as demonstrated by the significant training time taken by techniques such as ADASYN, where training times of greater than 1600 min were recorded using a logistic regression classifier on a dataset with 19 classes. In contrast, the use of the synthetic dataset, which has a lower class disparity, has shown significantly more efficient experimentation, with training times reduced by up to 98% while still showing comparable performance trends. SMOTE+SMOTEEN was seen to be uniquely valuable, achieving exceptional performance improvements for AdaBoost, boosting its 6-class F1 score from 0.6979 to 0.9127 on actual data, and possessing exceptional computational efficiency on both datasets. Sample Weighting showed excellent preservation of performance for already-strong models post-balancing, making it an excellent choice for deployment scenarios. Random Forest was seen to be the strongest model regardless of dataset and balancing technique, making it an excellent candidate for IoMT security scenarios. Ultimately, this study proves synthetic data to be an excellent proxy for actual IoMT scenarios, allowing for rapid, resource-friendly testing without the limitations imposed by imbalanced data sets. This study presents an excellent strategic approach for choosing balancing techniques, SMOTE+SMOTEEN for achieving optimal performance and efficiency, Sample Weighting for preserving high-tier performance, and synthetic data for rapid development. By choosing balancing techniques that align with system needs and resource availability, it is possible to create stronger, more efficient, and deployable IDSs for this critical IoMT world.

Acknowledgement: The project was funded by KAU Endowment (WAQF) at King Abdulaziz University, Jeddah, Saudi Arabia. The authors, therefore, acknowledge with thanks WAQF and the Deanship of Scientific Research (DSR) for technical and financial support. Also, they are grateful to the Center of Excellence in High Performance Computing for granting access to the "Aziz" Supercomputer, which was instrumental in conducting the experiments for this study.

Funding Statement: This research was funded by the KAU Endowment (WAQF) at King Abdulaziz University, Jeddah, Saudi Arabia. The authors, therefore, acknowledge with thanks the WAQF and the Deanship of Scientific Research (DSR) for their financial support. The APC was funded by number [RG-6-611-43].

Author Contributions: The authors confirm their contribution to the paper as follows: Study conception and design: Taghreed Alkhodaidi, Miada Almasre, Wadee Alhalabi. data collection: Taghreed Alkhodaidi. analysis and interpretation of results: Taghreed Alkhodaidi. draft manuscript preparation: Taghreed Alkhodaidi. All authors reviewed and approved the final version of the manuscript.

Availability of Data and Materials: The data that support the findings of this study are available from the corresponding author, [Taghreed Alkhodaidi], upon reasonable request.

Ethics Approval: The study did not require ethical approval.

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

References

1. Alanezi M, AL-Azzawi RMA. AI-powered cyber threats: a systematic review. Mesopotamian J CyberSecurity. 2024;4(3):166–88.
2. Aslan Ö, Aktuğ SS, Ozkan-Okay M, Yilmaz AA, Akin E. A comprehensive review of cyber security vulnerabilities, threats, attacks, and solutions. Electronics. 2023;12(6):1333. doi:10.3390/electronics12061333.
3. Nawaz MS, Raza MA, Raza B, Ahmad M, Syed F. AI-driven intrusion detection systems for securing IoT healthcare networks. Int J Adv Comput Sci Appl. 2025;16(6):489–97. doi:10.14569/ijacsa.2025.0160647.
4. Paramesha M, Rane NL, Rane J. Artificial intelligence, machine learning, and deep learning for cybersecurity solutions: a review of emerging technologies and applications. Part Univ Multidiscip Res J. 2024;1(2):84–109. doi:10.2139/ssrn.4855884.
5. Misbah A, Sebbar A, Hafidi I. Securing internet of medical things: an advanced federated learning approach. Int J Adv Comput Sci Appl. 2025;16(2):1305–16.
6. Hameed SS, Hassan WH, Latiff LA, Ghabban F. A systematic review of security and privacy issues in the internet of medical things; the role of machine learning approaches. PeerJ Comput Sci. 2021;7(4):e414. doi:10.7717/peerj-cs.414.
7. Zhang Y, Wang R, Wang Y, Chen M, Guizani M. Diversity-driven proactive caching for mobile networks. IEEE Trans Mob Comput. 2023;23(7):7878–94. doi:10.1109/tmc.2023.3340733.
8. Dilawar N, Rizwan M, Ahmad F, Akram S. Blockchain: securing internet of medical things (IoMT). Int J Adv Comput Sci Appl. 2019;10(1):82–9. doi:10.14569/ijacsa.2019.0100110.
9. Tauqeer H, Iqbal MM, Ali A, Zaman S, Chaudhry MU. Cyberattacks detection in IoMT using machine learning techniques. J Comput Biomed Inform. 2022;4(01):13–20. doi:10.56979/401/2022/80.
10. Kulshrestha P, Vijay Kumar T. Machine learning based intrusion detection system for IoMT. Int J Syst Assur Eng Manag. 2024;15(5):1802–14. doi:10.1007/s13198-023-02119-4.
11. Binbusayyis A, Alaskar H, Vaiyapuri T, Dinesh M. An investigation and comparison of machine learning approaches for intrusion detection in IoMT network. J Supercomput. 2022;78(15):17403–22. doi:10.1007/s11227-022-04568-3.
12. Zhou ZH. Machine learning. Cham, Switzerland: Springer Nature; 2021.

13. Dadkhah S, Mahdikhani H, Danso PK, Zohourian A, Truong KA, Ghorbani AA. Towards the development of a realistic multidimensional IoT profiling dataset. In: 2022 19th Annual International Conference on Privacy, Security & Trust (PST). Piscataway, NJ, USA: IEEE; 2022. p. 1–11.
14. Ramesh K, Miller NC, Faridi A, Aloul F, Zualkernan I, Sajun AR. Efficient machine learning frameworks for strengthening cybersecurity in internet of medical things (IoMT) ecosystems. In: 2024 IEEE International Conference on Internet of Things and Intelligence Systems (IoTaIS). Piscataway, NJ, USA: IEEE; 2024. p. 92–8.
15. Chandekar P, Mehta M, Chandan S. Enhanced anomaly detection in IoMT networks using ensemble AI models on the CICIoMT2024 dataset. arXiv:2502.11854. 2025.
16. Alsbatin L, Alrifai BM, Zawaideh F, Alawneh TA. Advancing IoMT security: machine learning-based detection and classification of multi-protocol cyberattacks. J Wirel Mob Netw Ubiquitous Comput Dependable Appl. 2025;16(2):228–47.
17. Salehpour A, Balafar MA, Souri A. An optimized intrusion detection system for resource-constrained IoMT environments: enhancing security through efficient feature selection and classification. J Supercomput. 2025;81(6):783. doi:10.1007/s11227-025-07253-3.
18. Doménech J, León O, Siddiqui MS, Pegueroles J. Evaluating and enhancing intrusion detection systems in IoMT: the importance of domain-specific datasets. Internet Things. 2025;32(1):101631. doi:10.1016/j.iot.2025.101631.
19. Mohsin M, Jony AI. A comparative analysis of medical IoT device attacks using machine learning models. Malays J Sci Adv Technol. 2024;4(4):429–39. doi:10.56532/mjsat.v4i4.318.
20. Goyal M, Kumar R. Machine learning for malware detection on balanced and imbalanced datasets. In: 2020 International Conference on Decision Aid Sciences and Application (DASA). Piscataway, NJ, USA: IEEE; 2020. p. 867–71.
21. Susan S, Kumar A. The balancing trick: optimized sampling of imbalanced datasets—a brief survey of the recent state of the art. Eng Rep. 2021;3(4):e12298. doi:10.1002/eng2.12298.
22. Altalhan M, Algarni A, Alouane MTH. Imbalanced data problem in machine learning: a review. IEEE Access. 2025;13(1):13686–99. doi:10.1109/access.2025.3531662.
23. Shah Mirkhail A, Zhang X. Deep learning for anomaly detection in IoT healthcare systems. Int Res J Multidiscip Scope. 2025;6(2):1480–94. doi:10.47857/irjms.2025.v06i02.03768.
24. Areia J, Bispo IA, Santos L, Costa RLDC. IoMT-TrafficData: dataset and tools for benchmarking intrusion detection in internet of medical things. IEEE Access. 2024;12:115370–85.
25. Rehman M, Kalakoti R, Bahşi H. Comprehensive feature selection for machine learning-based intrusion detection in healthcare IoMT networks. In: 11th International Conference on Information Systems Security and Privacy. Setúbal, Portugal: SCITEPRESS; 2025. p. 248–59.
26. Yazdinejad A, Karimipour H, Halabi T. Toward stress-adaptive cyber defense: cognitive-physiological synchronization in IoT environments. IEEE Internet Things J. 2026;13(7):13832–48.
27. Wahab SA, Sultana S, Tariq N, Mujahid M, Khan JA, Mylonas A. A multi-class intrusion detection system for DDoS attacks in IoT networks using deep learning and transformers. Sensors. 2025;25(15):4845. doi:10.3390/s25154845.
28. AlFuraih D, Mhamdi L, Karar AS. Explainable hybrid CNN-XGBoost framework for multi-class IoT intrusion detection with leakage-aware feature selection. Appl Syst Innov. 2026;9(3):49.
29. Jaiswal R, Andersen PA, Cenkeramaddi LR, Jiao L, Granmo OC. A tsetlin machine-driven intrusion detection system for next-generation IoMT security. arXiv:2604.03205. 2026.
30. Alzubi JA, Alzubi OA, Qiqieh I, Singh A. A blended deep learning intrusion detection framework for consumable edge-centric iomt industry. IEEE Trans Consum Electron. 2024;70(1):2049–57. doi:10.1109/tce.2024.3350231.
31. Dadkhah S, Neto ECP, Ferreira R, Molokwu RC, Sadeghi S, Ghorbani A. CICIoMT2024: attack vectors in healthcare devices-a multi-protocol dataset for assessing IoMT device security. Preprints. 2024. doi:10.20944/preprints202402.0898.v1.
32. Lin J. Divergence measures based on the Shannon entropy. IEEE Trans Inf Theory. 2002;37(1):145–51. doi:10.1109/18.61115.