International Journal of Safety and Security Engineering
Vol. 16, No. 10, October, 2026, pp. 2307-2321
Journal homepage: http://iieta.org/journals/ijsse

AI-Driven Threat Detection in the Internet of Medical Things: An Imbalance-Aware Deep and Ensemble Learning Framework with Domain-Informed Feature Engineering

Mohammed T. Abdulhadi*, Wisam Ali Mahmood, Anmar A. Al-Janabi, Ayad R. Abbas

College of Computer Science, University of Technology, Baghdad 10066, Iraq

Corresponding Author Email: mohammed.t.abdulhadi@uotechnology.edu.iq

Copyright: ©2026 The authors. This article is published by IIETA and is licensed under the CC BY 4.0 license (http://creativecommons.org/licenses/by/4.0/).

https://doi.org/10.18280/ijsse.161010

Received: 31 August 2026
Revised: 29 September 2026
Accepted: 9 October 2026
Available online: 31 October 2026

Keywords:
AI, class imbalance, deep learning, ensemble learning, imbalance-aware learning, Internet of Medical Things, SMOTE-Tomek, threat detection

ABSTRACT

AI has become the primary means of detecting threats in the Internet of Medical Things (IoMT), where connected clinical devices generate continuous network and physiological data streams beyond the reach of static, rule-based defences. Class imbalance, however, distorts the learning problem: benign traffic dominates the data, so models optimised for overall accuracy under-detect the minority attack class. This paper presents an imbalance-aware AI framework for IoMT threat detection, evaluated on the WUSTL-EHMS-2020 dataset using a binary threat target defined as the top 30% of a composite risk score derived from network and physiological indicators. A domain-informed representation stage derives 32 features by combining multi-sensitivity statistical anomaly analysis of network flows, clinically grounded physiological thresholding and port-based exposure profiling. Seven resampling strategies (Synthetic Minority Oversampling Technique (SMOTE), Adaptive Synthetic Sampling (ADASYN), Borderline-SMOTE, Support Vector Machine SMOTE (SVM-SMOTE), KMeans-SMOTE, SMOTE-Tomek and SMOTE-Edited Nearest Neighbors (ENN)) are compared under an identical protocol and paired with conventional and internally balanced ensemble learners and a class-weighted deep neural network. SMOTE-Tomek yields the most learnable representation (F1-score = 0.937, Area Under the Receiver Operating Characteristic Curve (AUC) = 0.973), whereas ADASYN degrades performance (F1-score = 0.776) by injecting synthetic noise into overlapping regions. Balanced ensemble learners reach recall above 0.94 against 0.654–0.800 for their unbalanced counterparts, and the class-weighted deep network attains an F1-score of 0.923 and an AUC of 0.970. Coupling data-level resampling with algorithm-level balancing, a double-balancing effect, is the dominant driver of detection gains in AI-based IoMT security.

1. INTRODUCTION

Internet of Medical Things (IoMT) devices have become an essential part of the current health care ecosystem, comprising a wide variety of connected medical devices such as patient monitors, infusion pumps, wearable biosensors, and diagnostic imaging tools. The global IoMT market was valued at approximately USD 230.7 billion in 2024 and is projected to reach USD 658.6 billion by 2030, corresponding to a compound annual growth rate of 18.2% over 2025–2030 [1]. Such devices collect a large volume of private patient information, which allows for continuous monitoring and prediction-based diagnosis and treatment options [2]. However, the growing use of IoMT devices also raises real security and privacy concerns [3, 4].

The healthcare industry has witnessed a significant rise in the number of cyber-attacks, wherein IoMT devices have become a key entry point for network attacks and the theft and extortion of sensitive data. In just one year, i.e., 2023, there was an estimated occurrence of 725 health data breaches reported to the U.S. Department of Health and Human Services [5]. The impact of successful cyber-attacks against medical devices is more than a data breach; it could be life-threatening due to the tampering of the devices or fake measurements of sensors, causing denial of access to such services [6, 7]. It is evident that there is an immediate need for developing cybersecurity frameworks that consider the peculiarities of medical devices [8, 9].

A key issue in designing efficient intrusion detection systems (IDSs) for IoMT ecosystems is the class imbalance that occurs in network security datasets. The amount of malicious activity present in normal network traffic is minimal; therefore, datasets used in machine learning algorithms will have far more normal traffic instances than attacks [10, 11]. The severity of the effect varies widely across corpora, from ratios of 100:1 or worse in operational traffic captures down to the moderate 2.33:1 of the score-defined target used here, and the present study is deliberately positioned at the conservative end of that range. Machine learning classifiers trained using unbalanced datasets are known to be very accurate overall yet unable to accurately predict the minority class instances, which is precisely what they should be designed to detect [12, 13]. Unbalanced dataset training could lead to an unacceptably high percentage of false

negatives, where actual attacks will not be detected at all [14, 15].

To overcome such difficulties, many studies in the field have introduced several approaches at the data level and algorithm level. The methods that have proven their efficacy include Synthetic Minority Oversampling Technique (SMOTE), as well as several of its variations such as Adaptive Synthetic Sampling (ADASYN), Borderline-SMOTE, and hybrids such as SMOTE-Tomek. Despite their proven efficacy, their comparative evaluation within the IoMT cybersecurity context is not sufficiently examined [16-19]. In addition, the use of balanced ensembles along with external balance techniques such as the ones offered by BalancedRandomForestClassifier and EasyEnsembleClassifier has not been sufficiently analyzed for IoMT threats yet [20, 21].

This work seeks to fill these gaps through the following contributions: (1) We propose a comprehensive multilayer security architecture that incorporates network traffic analysis, medical device anomaly detection, and threat composite scoring that is tailored for the IoMT domain, producing 32 security attributes. (2) We systematically evaluate seven state-of-the-art data balancing methods and measure their effect on threat detection accuracy. (3) We prove that the use of SMOTE-Tomek balancing along with ensemble classifiers results in excellent threat detection, with improved F1-scores rising from 0.772–0.777 for imbalanced data sets to 0.934–0.938 for balanced data sets. (4) The efficacy of our approach is demonstrated using the WUSTL-EHMS-2020 IoMT Security Dataset consisting of 16,318 data instances with 45 attributes [22].

2. RELATED WORK

Research on machine-learning threat detection for the IoMT converges on three questions that bear directly on the present study: how far flow-level and physiological signals can be fused into a single detector, how the class distribution of security data is treated before training, and how much of the reported performance survives once that distribution is taken into account. The literature is therefore reviewed along those three lines rather than as a sequence of individual studies.

2.1 Machine learning detection on the WUSTL-EHMS-2020 testbed

The WUSTL-EHMS-2020 dataset was introduced together with a fused detector, in which network-flow attributes and patient biometrics were combined into a single feature vector and evaluated with Support Vector Machine (SVM), Random Forest (RF), k-nearest neighbours (kNN) and an artificial neural network (ANN), the last attaining 90.04% accuracy; the fused representation outperformed either view in isolation, establishing the premise on which subsequent work rests [22]. Later studies largely retain that premise while substituting stronger learners. A comparison of decision tree, RF, Naive Bayes and SVM under several feature-selection schemes placed RF highest at 97.86% [23]; a kernel pipeline combining kernel principal component analysis (KPCA) and kernel partial least squares (KPLS) with a kernel extreme learning machine (KELM) classifier reported 99.9% accuracy at 100% sensitivity [24]; and an eXtreme Gradient Boosting (XGBoost)-SVM ensemble reached 98.04% on the same data, with additional validation on TON-IoT and CICIDS-2017 [25].

These accuracies are not, however, mutually comparable, because none of the studies alters the 87.5:12.5 class ratio of the original labels before training and all report performance on it. Accuracy computed on such a distribution is dominated by the benign class and can remain high while a substantial proportion of attack flows pass undetected, so the minority recall that governs clinical risk is left unquantified [22, 23]. The near-perfect figures [24] sharpen this concern rather than resolve it: when preprocessing leakage is explicitly controlled, a deep sequential model on the same testbed settles at 99% using 25 of the 45 attributes, and the inflation reported elsewhere on this dataset is attributed in part to transformations fitted before the train/test partition [26]. A systematic study of the interaction between network architecture, dimensionality reduction and imbalance correction on WUSTL-EHMS-2020 reaches a compatible conclusion, finding the combination of preprocessing steps to affect the outcome as strongly as the choice of classifier, with unsuitable combinations actively degrading detection [27].

2.2 Treatment of class imbalance in security data

Where imbalance is addressed at all, the treatment is typically a single resampler applied once and accepted without comparison. SMOTE combined with a long short-term memory (LSTM) detector improved minority-class detection appreciably, but was validated only on NSL-KDD, a general-purpose intrusion corpus containing no healthcare-specific traffic [28]. The same resampler, paired with Pearson-correlation feature selection and an RF, produced the most favourable precision-recall trade-off in an Internet of Things (IoT) healthcare anomaly framework [29]. A hybrid of an adaptively trained C4.5 tree and a deep Q-network went one step further by correcting the distribution at the data level with SMOTE and the attention of the classifier at the algorithm level with a focal loss simultaneously, reporting 99.03% accuracy on CICIoMT-2024 [30]. Survey work confirms that this is the general pattern: classifications of machine-learning-based IoMT intrusion detection into signature, anomaly and hybrid approaches note the advantage of anomaly methods against previously unseen attacks and identify class imbalance, alongside feature selection and computational cost, as the open problems of the field [31]. What is consistently absent is controlled comparison; the preference for SMOTE over ADASYN, Borderline-SMOTE or a hybrid cleaning variant is rarely justified against those alternatives on the same data under the same classifier.

2.3 Algorithm-level correction and ensemble learning

A second line of work corrects the imbalance inside the learner rather than in the data. Ensembles of RF, gradient boosting, AdaBoost and voting classifiers consistently outperform their single-classifier counterparts on IoMT traffic, yet class imbalance persists as the dominant source of residual error on the rarer attack classes even after ensembling, which indicates that aggregation alone does not compensate for a skewed training prior [32]. Feature selection has been used in a similar way: one anomaly-based IoMT IDS applied recursive feature elimination with a Ridge penalty before a mix of conventional and deep classifiers [33]. Both types of correction, data-level resampling and algorithm-level

weighting or balanced ensembling, are therefore well represented, but they are almost always evaluated separately.

2.4 Research gap

Four gaps follow from the above. First, comparative evidence is missing: studies commit to one or two balancing strategies without evaluating the wider family under identical conditions, so the reported choice cannot be separated from the classifier it happened to be paired with [28-30]. Second, external and internal correction are treated as alternatives rather than as methods that work together, and their combination has not been tested [32, 33]. Third, the gain from balancing is rarely measured against the same model trained without balancing and tested on the same held-out data, which is needed to separate the effect of the correction from that of the model [22, 23, 25]. Fourth, no study on this dataset builds features from clinical knowledge of the monitored vital signs, even though the biometric data is exactly what makes IoMT traffic different from ordinary network traffic [22, 26]. This work addresses all four points.

3. PROPOSED METHODOLOGY

3.1 System architecture overview

The suggested Advanced Balanced Cybersecurity IoMT System follows a seven-phase analysis procedure, which is shown in Figure 1.



Figure 1. Proposed system architecture—seven-phase analysis pipeline
Note: Secure Hash Algorithm 256-bit (SHA-256); imbalance ratio (IR); Synthetic Minority Oversampling Technique (SMOTE).

The first phase, Secure Data Loading, loads the dataset and checks file integrity with Secure Hash Algorithm 256-bit (SHA-256) hashes. The second phase, Imbalance Analysis, examines the class distribution in detail. In Phase 3, Balancing Application, the seven balancing methods are applied and compared. Phase 4 (Baseline Training) trains the base models on the imbalanced data. In Phase 5 (Balanced Training), the models are trained on the resampled data using balanced ensemble classifiers. Phase 6 (Dashboard Generation) produces interactive security dashboards.

The pipeline is linear on purpose rather than end-to-end. Each phase saves its intermediate output to disk, so the effect of a single choice, such as a threshold, a resampler or a class weight, can be checked without re-running the whole pipeline.

Phase 7, Reporting and Persistence, is the last step. It saves the trained models, the scalers and the resampling settings in one versioned bundle, together with the file hashes from Phase 1.

One design decision governs everything that follows, namely the position of the resampling operator relative to the train/test partition. If resampling is carried out on the complete dataset before splitting, a synthetic training point may be an interpolation between two records that later land in the test partition; the partition is then no longer independent, and every score computed on it is optimistic by an amount that cannot be estimated after the fact. The partition is therefore performed first. Each resampler is fitted on the training portion alone and is never applied to the held-out records, which retain the original class proportions of the dataset. All models, balanced and unbalanced alike, are consequently scored against the same untouched distribution, and the comparison between the unbalanced and balanced models measures the effect of the training procedure rather than the effect of changing the evaluation set.

3.2 Data preparation and integrity control

Before any modelling, the raw comma-separated values (CSV) file is checked against a stored SHA-256 digest. This protects against file corruption during transfer and against accidentally using a partly preprocessed copy, a common and easily missed cause of results that cannot be reproduced in intrusion-detection work.

Four preparation steps follow. First, exact duplicate records are removed, because duplicates that end up in both partitions raise the measured accuracy without adding any information. Second, the source Media Access Control (MAC) address field is dropped. This field does not describe behaviour; it only identifies the machine that sent the flow. Since the WUSTL-EHMS-2020 labels were assigned from the attacker laptop's MAC address, keeping it would let a classifier read the label directly and make the whole experiment meaningless. Third, records with missing biometric readings are imputed with the median of the corresponding column computed on the training partition only, the median being preferred to the mean because several physiological channels are visibly skewed by sensor dropout. Fourth, all continuous attributes are standardised with a RobustScaler, which centres on the median and scales by the interquartile range (IQR). A standard z-score transform was rejected here because the network-flow features contain heavy-tailed outliers that are themselves informative; a mean-based scaler allows those outliers to compress the useful range of the remaining observations.

The binary target used in all experiments is not the MAC-derived label distributed with WUSTL-EHMS-2020 (14,272 normal and 2,046 attack records). It was defined from the composite threat score S of Eq. (7): records whose score exceeds the 70th percentile of S over all 16,318 records are

labelled threat (4,896 records, 30.0%), and the remainder normal (11,422 records, 70.0%). No records were removed or added. Because the components of S are also among the model inputs, the reported results measure how well each classifier recovers this score-defined risk target; they are not detection rates against the attacks labelled in the original release.

The scaler, like every other transformation with learnable state, is fitted on the training partition and merely applied to the test partition. The partition itself is stratified on the class label with a 70/30 ratio and a fixed seed, so that the class proportions of the full dataset are preserved on both sides of the split. The resulting training partition contains 11,422 records (7,995 normal and 3,427 threat), and the test partition contains 4,896 records (3,427 normal and 1,469 threat).

3.3 Feature engineering pipeline

The feature engineering pipeline transforms the raw 45-feature WUSTL-EHMS-2020 dataset into 32 enhanced security features through four complementary analysis modules, as shown in Figure 2.

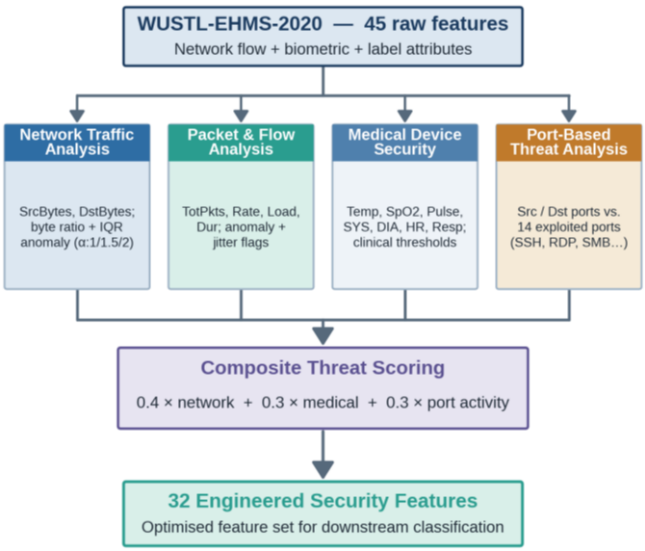

Figure 2. Feature engineering pipeline – four analysis modules

Note: interquartile range (IQR); body temperature (Temp); peripheral oxygen saturation (SpO_2); systolic blood pressure (SYS); diastolic blood pressure (DIA); heart rate (HR); respiratory rate (Resp); source (Src); destination (Dst); Secure Shell (SSH); Remote Desktop Protocol (RDP); Server Message Block (SMB).

Network Traffic Analysis Module: This module analyzes SrcBytes and DstBytes to calculate the bytes ratio and total bytes transferred. The multi-sensitivity anomaly detection method uses IQR-based anomaly detection for three different sensitivities: α = 1.0 for strict, α = 1.5 for moderate, and α = 2.0 for loose. If samples fall outside the range of $[Q_1 - \alpha \times IQR, Q_3 + \alpha \times IQR]$, then that sample will be considered an anomaly for that particular sensitivity type. These three anomalies are then calculated together to create a weighted sum of bytes_anomaly_combined = 3 × strict + 2 × moderate + 1 × loose.

Packet and Flow Analysis Module: Anomaly detection is carried out based on the TotPkts, Rate, Load, and Dur parameters. In each case, the anomaly score and high-value flags are determined based on the IQR score (α = 1.0–1.2) and the percentile score (85th–90th percentile). The source and destination jitters are aggregated into a single jitter score, with respective flags.

Medical Device Security Module: The following seven physiological parameters will be considered—body temperature (Temp), peripheral oxygen saturation (SpO_2), Pulse_Rate, systolic blood pressure (SYS), diastolic blood pressure (DIA), Heart_Rate, and respiratory rate (Resp_Rate). Critical thresholds for alerts to be raised due to clinical knowledge will be defined for the following conditions—temperature >39 ℃ or < 35 ℃, SpO_2 < 88%, and heart rate > 130 or < 50 beats per minute.

Port-Based Threat Analysis Module: Source and destination ports are evaluated against 14 commonly exploited ports: 21 for File Transfer Protocol (FTP), 22 for Secure Shell (SSH), 23 for Telnet, 80 for Hypertext Transfer Protocol (HTTP), 135 for Remote Procedure Call (RPC), 139 for Network Basic Input/Output System (NetBIOS), 443 for Hypertext Transfer Protocol Secure (HTTPS), 445 for Server Message Block (SMB), 993 for Internet Message Access Protocol Secure (IMAPS), 995 for Post Office Protocol 3 Secure (POP3S), 1433 for Microsoft SQL Server (MSSQL), 3389 for Remote Desktop Protocol (RDP), 5900 for Virtual Network Computing (VNC), and 6667 for Internet Relay Chat (IRC).

Composite Scoring: The final composite_threat_score integrates all modules: 0.4 × network_risk_score + 0.3 × medical_risk_score + 0.3 × suspicious_port_activity.

The four modules described above are formalised as follows. For a flow with SrcBytes and DstBytes, the directional ratio and the transferred volume are

$$r^b = \frac{SrcBytes}{DstBytes + \varepsilon}, \quad b^{tot} = SrcBytes + DstBytes \tag{1}$$

where, $\varepsilon = 1$ prevents division by zero on unidirectional flows. A pronounced asymmetry in r^b is characteristic of injection traffic, in which the compromised gateway emits far more than it receives.

Statistical anomaly detection is applied to each continuous attribute through the IQR. With Q_1 and Q_3 the first and third quartiles and $IQR = Q_3 - Q_1$, the acceptance interval at sensitivity α is

$$L_\alpha = Q_1 - \alpha \times IQR, \quad U_\alpha = Q_3 + \alpha \times IQR \tag{2}$$

and the corresponding indicator for an observation v is

$$a_\alpha(v) = 1 \text{ if } v < L_\alpha \text{ or } v > U_\alpha, \text{and 0 otherwise} \tag{3}$$

Rather than committing to a single sensitivity, three are evaluated in parallel and combined into a graded score, so that a value flagged only by the permissive threshold is not treated as equivalent to a value flagged by all three:

$$A^b = 3 \cdot a_{1.0}(v) + 2 \cdot a_{1.5}(v) + 1 \cdot a_{2.0}(v), \quad A^b \in [0, 6] \tag{4}$$

The weights 3, 2 and 1 require no calibration, and the ordering they induce is invariant to their particular values. The three acceptance intervals are nested by construction, since the interval at α = 1.0 is contained in that at α = 1.5, which is in turn contained in that at α = 2.0. An observation lying outside the widest interval therefore lies outside the two narrower ones

as well, so the indicator vector $(a_{1.0}, a_{1.5}, a_{2.0})$ admits only four states—(0,0,0), (1,0,0), (1,1,0) and (1,1,1)—and A^b attains only the four values 0, 3, 5 and 6, which rank an observation by how far outside the acceptance region it falls. Any strictly positive weighting yields four values in that same order. The weights are therefore a labelling of an ordinal scale and not a tunable parameter of the model.

The medical module operates on a different logic. Physiological limits are not learned from the data but taken from routine clinical practice, since a reading that is implausible for a living patient is suspicious regardless of how frequently it appears in the dataset. Writing N for the set of monitored channels and $[l_j, u_j]$ for the admissible range of channel j, the medical risk of a record is the normalised count of violations,

$$R^m = (1/|N|) \cdot \Sigma_j 1[x_j < l_j \text{ or } x_j > u_j] \quad (5)$$

The port module contributes a third component. For source and destination ports p_s and p_d and the set P of fourteen frequently exploited service ports, the exposure term is

$$R^P = ½ \cdot (1[p_s \in P] + 1[p_d \in P]) \quad (6)$$

The three components are finally aggregated into a single composite indicator,

$$S = 0.4 \cdot R^n + 0.3 \cdot R^m + 0.3 \cdot R^P \quad (7)$$

where, R^n is the normalised network risk obtained from the byte, packet and jitter anomaly scores. The larger weight on the network component reflects the fact that the man-in-the-middle attacks recorded in the testbed manifest first at the transport layer, and only afterwards in the biometric stream. S enters the model as an engineered feature and also defines the classification target.

Because S also defines the classification target, the choice of 0.4, 0.3 and 0.3 determines which records are labelled threat. Each of its three constituents enters the feature set in its own right: R^n is retained as network_risk_score, R^P as suspicious_port_activity, and R^m is the mean of the seven physiological alert flags, every one of which is itself a feature. A learner can therefore reconstruct the target directly from these constituents, which is why the reported results measure agreement with this risk definition rather than detection of the originally labelled attacks. The particular values assign the larger share to the network component for the reason given above and divide the remainder equally between the physiological and port channels, neither of which is given precedence over the other.

The four modules together produce 32 features, comprising eleven flow-derived quantities, nine graded anomaly indicators, seven physiological alert flags, three port-exposure terms and two aggregate risk scores. The 70th-percentile threshold that defines the target is computed over all 16,318 records before partitioning.

3.4 Data balancing techniques

Seven different balance techniques are analyzed: SMOTE [16] creates new examples of the minority class through interpolation (15,990, 1:1); ADASYN [34] creates more examples in complex areas (15,631, approximately 1:1); Borderline-SMOTE [35] considers only examples located at the borders (15,990, 1:1); Support Vector Machine SMOTE (SVM-SMOTE) relies on SVM decision borders (15,990, 1:1); KMeans-SMOTE uses clustering and then applies SMOTE (15,756, approximately 1:1); SMOTE-Tomek [17] is a combination of SMOTE with removal of Tomek links (14,556, 1:1); SMOTE-Edited Nearest Neighbors (ENN) combines SMOTE with ENN [36] (12,921). An automatic selection algorithm recommends the optimal technique based on the imbalance ratio.

The seven strategies differ less in mechanism than in where they place synthetic points relative to the decision boundary, and that distinction turns out to matter more than the amount of oversampling performed. SMOTE interpolates uniformly along the segment joining a minority point to one of its k nearest minority neighbours, treating all regions of the minority manifold alike. ADASYN shifts the generation density towards points whose neighbourhoods are dominated by the majority class, on the reasoning that these are the hardest cases; the difficulty is that in a dataset with overlapping classes those same neighbourhoods are precisely where labelling noise concentrates, so the method amplifies the least reliable part of the training signal. Borderline-SMOTE adopts a middle position, restricting generation to minority points that have at least one majority neighbour but are not entirely surrounded by them. SVM-SMOTE takes the support vectors of a fitted SVM as generation seeds, and KMeans-SMOTE first clusters the feature space and oversamples within clusters that are already minority-dense, which limits the creation of synthetic points in regions where the minority class has no genuine presence.

The two hybrid methods add a cleaning stage after oversampling. A Tomek link is a pair of nearest neighbours belonging to opposite classes; such pairs lie by construction on the class boundary and are either noise or borderline cases. SMOTE-Tomek removes them after oversampling, which sharpens the boundary that the classifier subsequently has to learn and, incidentally, yields a resampled set smaller than that produced by plain SMOTE. SMOTE-ENN applies the Edited Nearest Neighbours rule instead, discarding any point whose label disagrees with the majority of its three nearest neighbours. ENN is the more aggressive of the two filters and removes genuine minority observations along with the noisy ones, which is the reason its resampled set is the smallest of the seven.

All resamplers use k = 5 neighbours and a fixed random seed, and all target a 1:1 ratio except SMOTE-ENN and ADASYN, whose cleaning and density-adaptive steps respectively make the final ratio a consequence of the data rather than a parameter. Because resampling is confined to the training partition, the reported sample counts refer to the resampled training set only and not to the dataset as a whole.

3.5 Deep neural network architecture

The improved deep neural network model includes seven dense layers with batch normalization, dropout regularization (0.2–0.4), Adam optimizer (learning rate = 0.001, decay = 1×10^{-6}), class-weighted binary cross-entropy loss function, early stopping (patience = 25), and reduce learning rate on plateau. Figure 3 shows the improved deep neural network model that is adopted for the binary classification problem. The deep neural network model takes input in the form of a vector containing 32 features and processes the input data through a series of fully connected dense layers of neurons whose size decreases as follows: 1024, 512, 256, 128, 64, 32, and 16

neurons. The early dense layers of neurons have Batch Normalization layers for stability in learning, whereas Dropout is used to counter overfitting problems. The last dense layer consists of a Sigmoid activation function layer, which gives binary probability outputs for the input data. This improved model is optimized using Adam with a learning rate of 0.001, the binary cross-entropy loss function, and Early Stopping.

The width schedule was chosen so that each layer roughly halves the representation, which keeps the number of parameters concentrated in the first two layers where the 32 input features are recombined, and leaves the deeper layers to refine rather than expand the representation. Batch normalisation is placed after the activation in the first four blocks only; in the narrow tail layers, it was found to add instability without measurable benefit. Dropout is graded from 0.4 in the widest block down to 0.2 in the narrowest.

Training uses the Adam optimiser with an initial learning rate of 0.001 and a decay of 1×10^{-6}, a batch size of 256, and a maximum of 200 epochs. Two callbacks govern termination. Early stopping monitors validation loss with a patience of 25 epochs and restores the weights of the best epoch rather than the last. Learning-rate reduction on plateau halves the rate after ten stagnant epochs, with a floor of 1×10^{-6}. 15% of the training partition is held out as a validation split for these callbacks; the test partition is never consulted during training.

The loss is a class-weighted binary cross-entropy in which the weight of each class is inversely proportional to its frequency in the training partition. This is an algorithm-level correction and is deliberately retained even when the training data have already been resampled, since the two mechanisms act at different points: resampling alters the empirical distribution the network sees, whereas the weighting alters the cost the network pays for each error. Their combination is what is referred to later as double balancing.

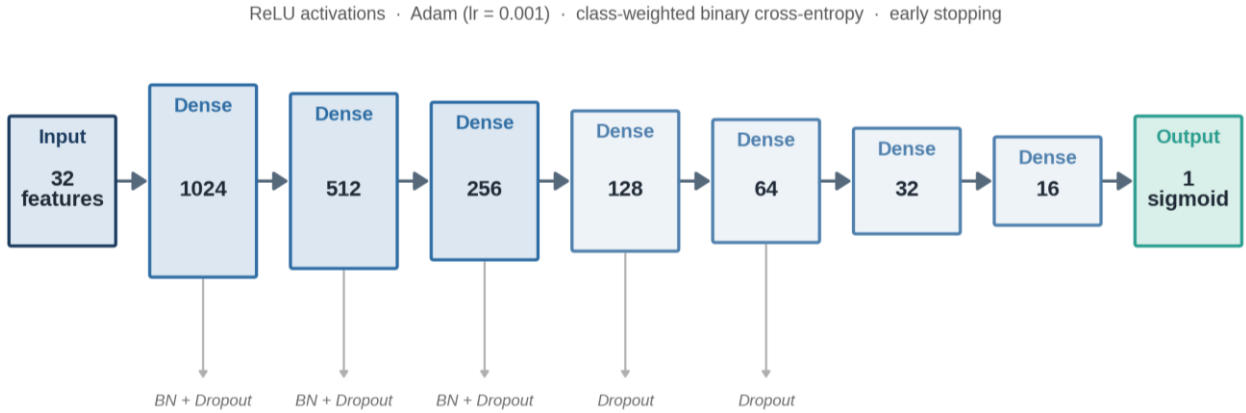

Figure 3. Enhanced deep neural network architecture
Note: Rectified Linear Unit (ReLU); learning rate (lr); Batch Normalization (BN).

3.6 Classification models and training protocol

Baseline Models: (1) RF Ensemble (300 trees, max_depth = 20, balanced class weights); (2) Gradient Boosting (200 estimators, learning rate = 0.08, max_depth = 10); (3) Deep Multilayer Perceptron (MLP) with 512, 256, 128 and 64 units, Rectified Linear Unit (ReLU) activation and RobustScaler input scaling.

Balanced Models: (1) BalancedRandomForestClassifier (300 estimators); (2) BalancedBaggingClassifier (200 estimators); (3) EasyEnsembleClassifier (100 estimators); (4) Weighted RF; (5) Gradient Boosting; (6) MLP; (7) Enhanced Deep Neural Network. All experiments use stratified 70/30 splits with random_state = 42.

Hyperparameters were fixed a priori from values commonly used for tabular intrusion-detection problems and were not tuned per configuration. This is a conservative choice. Tuning each model against the balanced training sets would very likely raise the reported figures, but it would also confound the effect of resampling with the effect of the search budget, and the question under study here is the former.

Classifier used for the resampling comparison: the seven balancing techniques compared in Section 4.2 are evaluated with one classifier held fixed across all seven configurations, namely the RF ensemble described above with 300 trees and a maximum depth of 20, trained with identical hyperparameters and an identical random seed on each resampled training set and evaluated on the same held-out partition. This is a necessary clarification, because a resampling method does not itself generate predictions and therefore has no F1-score, recall, precision or Area Under the Receiver Operating Characteristic Curve (AUC) of its own; the values in Section 4.2 are the scores of that fixed classifier trained on the training set produced by each method. Holding the learner constant is what permits the differences in Section 4.2 to be attributed to the resampler rather than to the model.

3.7 Evaluation metrics and experimental environment

Accuracy is reported for comparability with earlier work but is not used as a selection criterion. On a dataset in which the majority class dominates, a degenerate classifier that never raises an alert already achieves a high accuracy while detecting nothing, so the figure carries little information about the behaviour that matters. The four metrics used for evaluation are therefore defined on the confusion matrix of the attack class, with true positives (TP), false positives (FP), true negatives (TN) and false negatives (FN) counted with the attack class as positive:

$$Precision = \frac{TP}{TP + FP} \tag{8}$$

$$Recall = \frac{TP}{TP + FN} \tag{9}$$

$$F1 = \frac{2 \cdot Precision \cdot Recall}{Precision + Recall} \quad (10)$$

and AUC, the area under the receiver operating characteristic curve, which summarises the ranking quality of the decision scores independently of the operating threshold and is therefore unaffected by the class prior.

Recall is treated as the primary criterion. The asymmetry is clinical rather than statistical: a false alarm on an infusion pump costs an engineer a few minutes of inspection, whereas an undetected data-injection attack on the same pump propagates falsified vital signs into the record a clinician acts upon. Precision is nevertheless monitored, because an alerting system that is routinely wrong is switched off by its operators, and a detector that has been disabled has a recall of zero in practice, whatever it achieves on a benchmark.

Experiments were implemented in Python 3.10 with scikit-learn, imbalanced-learn 0.11 and TensorFlow/Keras. The random seed was fixed at 42 for the partition, the resamplers and the model initialisers, and the seed was not varied across the reported configurations.

3.8 Dataset description

WUSTL-EHMS-2020 dataset [22] was collected at Washington University in St. Louis from an active testbed of an Enhanced Healthcare Monitoring System (EHMS) that involves biomedical sensors, a gateway, networking systems, and a control server. Data were acquired through the use of the Audit Record Generation and Utilization System (ARGUS).

This dataset includes 16,318 records that have 45 attributes (35 network flows, 8 biometrics, 1 label, and 1 MAC). The label distribution in the dataset marks 2,046 records (12.5%) as attack and 14,272 (87.5%) as normal, whereas the score-defined target used in the experiments contains 4,896 threat (30.0%) and 11,422 normal (70.0%) records. Both distributions are shown in Figure 4 and summarised in Table 1.

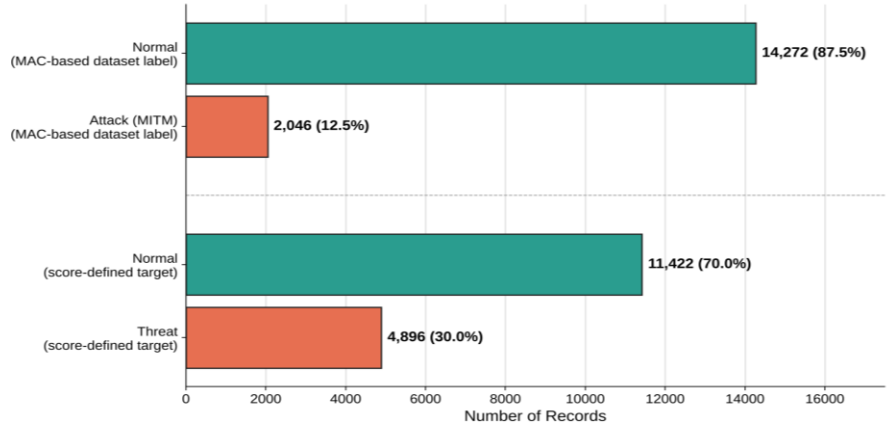

Figure 4. Class distribution of WUSTL-EHMS-2020 under the label distribution in the dataset and under the score-defined experimental target

Note: Media Access Control (MAC); Man-in-the-Middle (MITM). The experimental target labels a record as a threat when its composite threat score S of Eq. (7) exceeds the 70th percentile.

Table 1. WUSTL-EHMS-2020 dataset summary

Property	Value
Total Records	16,318
Features	45 (35 network + 8 biometric + 1 label + 1 MAC)
Normal / Threat (score-defined target)	11,422 (70.0%) / 4,896 (30.0%)
Imbalance Ratio (target)	2.33:1
Engineered Features	32
Attack Types	Man-in-the-middle (Spoofing, Data Injection)

Note: Media Access Control (MAC). The MAC-based label distributed with the dataset contains 14,272 normal (87.5%) and 2,046 attack (12.5%) records (imbalance ratio 6.98:1).

4. EXPERIMENTAL RESULTS

4.1 Class distribution after balancing

Figure 5 shows the class distribution before and after using the balancing methods. The score-defined target has an imbalance ratio of 2.33 to 1, where there are more samples in the normal category (70%) than in the threat category (30%). This is a moderate imbalance by the standards of intrusion-detection data, and it is described as such throughout: the target is not severely imbalanced, and the interest of the present comparison lies precisely in whether resampling still repays its cost at a ratio this mild. While both SMOTE-Tomek and SMOTE create a perfect 1-to-1 balance, they use different strategies to do so.

The difference between them lies in what happens after the synthetic points are placed. SMOTE stops once parity is reached, so the resampled training set contains every generated point, including those that fall inside the majority region. SMOTE-Tomek continues, identifying pairs of nearest neighbours from opposite classes and deleting them. The resampled set is consequently smaller than the one produced by SMOTE while remaining balanced, and the points that were removed are precisely those lying on the contested part of the boundary.

SMOTE-ENN behaves differently again. Its cleaning rule discards any observation whose label disagrees with the majority of its three nearest neighbours, and applies this to both classes rather than to boundary pairs alone. The resulting training set is the smallest of the seven (12,921 records), and inspection shows that the reduction falls disproportionately on genuine minority observations from sparse regions of the attack manifold. Aggressive cleaning of this kind removes noise and evidence at the same rate.

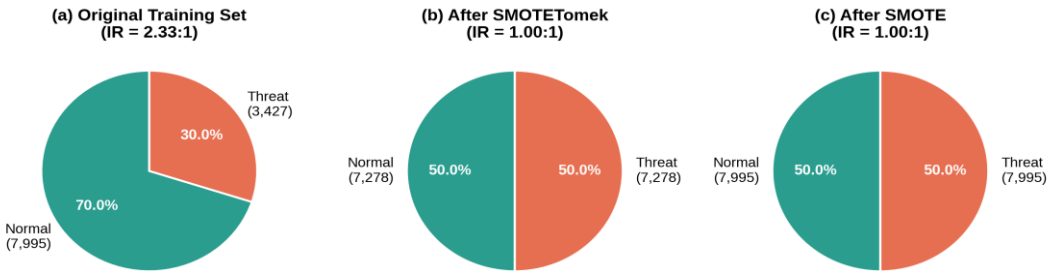

Figure 5. Class distribution of the training partition before and after balancing
Note: imbalance ratio (IR); Synthetic Minority Oversampling Technique (SMOTE).

4.2 Balancing techniques comparison

Comparative performance of the seven balancing methods is shown in Table 2 and Figure 6. SMOTE-Tomek produces the best results across all metrics (F1 = 0.937, AUC = 0.973), followed by SMOTE-ENN (F1 = 0.915, AUC = 0.958). The two boundary-informed oversamplers occupy the middle of the ranking, SVM-SMOTE at F1 = 0.883 and KMeans-SMOTE at F1 = 0.869, both ahead of uniform interpolation with plain SMOTE (F1 = 0.858) and of Borderline-SMOTE (F1 = 0.837). The worst performer is ADASYN (F1 = 0.776) owing to noise in the synthetic data created in highly overlapping regions.

The ordering of the seven techniques is stable across all reported metrics, which is itself informative: had the ranking depended on the metric, the differences could reasonably have been attributed to the operating threshold rather than to the quality of the resampled representation. The AUC column is the more diagnostic of the two, since it is threshold-free. SMOTE-Tomek reaches 0.973 against 0.869 for ADASYN, and a gap of that size in ranking quality cannot be recovered by any subsequent choice of decision threshold.

The failure of ADASYN admits a straightforward explanation. Its generation density is deliberately concentrated on minority points whose neighbourhoods are dominated by the majority class. In WUSTL-EHMS-2020, the two classes overlap substantially in the flow-derived features, because a spoofed flow is by design constructed to resemble a legitimate one, so the neighbourhoods ADASYN prioritises are the ones where class membership is genuinely ambiguous. The method therefore concentrates synthetic mass exactly where the training signal is least reliable, and the resulting F1 of 0.776 is lower than that of uniform SMOTE by a margin of 0.082.

Borderline-SMOTE sits between the two, which is consistent with this reading. It thus inherits part of ADASYN's exposure to overlap without inheriting all of it, and its F1 of 0.837 falls between the two accordingly.

SMOTE-Tomek removes only the boundary pairs left behind by oversampling and retains everything else, so its advantage over plain SMOTE is obtained with a smaller training set (14,556 against 15,990 records) and is therefore attributable to the removal of overlapping points rather than to the volume of synthetic data. SMOTE-ENN applies a stricter rule to both classes and reduces the training set to 12,921 records; it still ranks second at F1 = 0.915, but the margin it concedes to SMOTE-Tomek is consistent with the earlier observation that its cleaning discards genuine minority observations from sparse regions along with the noise. The two boundary-informed variants behave as their generation rules predict: SVM-SMOTE, which places synthetic points near an SVM-estimated decision boundary, and KMeans-SMOTE, which distributes them across clusters weighted by local minority density, both improve on uniform interpolation yet match neither hybrid, since neither performs any cleaning after generation.

Table 2. Performance comparison of the seven balancing techniques, each evaluated with the same Random Forest (RF) classifier (300 trees, max_depth = 20, identical random seed)

Technique	F1	Recall	Precision	AUC	Samples
SMOTE-Tomek	0.937	0.939	0.934	0.973	14,556
SMOTE-ENN	0.915	0.920	0.910	0.958	12,921
SVM-SMOTE	0.883	0.881	0.885	0.947	15,990
KMeans-SMOTE	0.869	0.867	0.871	0.941	15,756
SMOTE	0.858	0.856	0.860	0.936	15,990
Borderline-SMOTE	0.837	0.837	0.836	0.913	15,990
ADASYN	0.776	0.775	0.777	0.869	15,631

Note: Synthetic Minority Oversampling Technique (SMOTE); Edited Nearest Neighbors (ENN); Support Vector Machine (SVM); Adaptive Synthetic Sampling (ADASYN); Area Under the Receiver Operating Characteristic Curve (AUC).

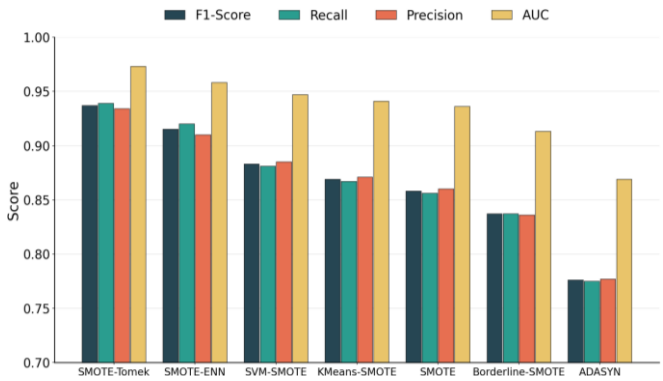

Figure 6. Performance comparison of data balancing techniques
Note: Synthetic Minority Oversampling Technique (SMOTE); Edited Nearest Neighbors (ENN); Support Vector Machine (SVM); Adaptive Synthetic Sampling (ADASYN); Area Under the Receiver Operating Characteristic Curve (AUC).

4.3 Original vs. balanced model performance

Tables 3 and 4 compare models trained on original imbalanced data against models trained on SMOTE-Tomek-balanced data. Figure 7 visualizes the comparison across all metrics.

Both tables are computed on the same held-out partition of 4,896 records (3,427 normal and 1,469 threat), which retains

the original class proportions; only the training procedure differs between them. The comparison therefore isolates the effect of the resampling and weighting decisions rather than reflecting a change in the evaluation distribution, a distinction that is easy to lose and that accounts for a number of implausibly large improvements reported elsewhere in this literature. Table 3 also supplies the simpler algorithm-level baseline against which resampling should be judged: the RF reported there is trained on the original, unresampled distribution with balanced class weights, and therefore represents class weighting applied alone. It reaches a recall of 0.800 and an F1 of 0.777, against 0.947 and 0.938 for the identically parameterised forest trained on the resampled data in Table 4. Class weighting on its own thus recovers only part of the available gain at this imbalance ratio.

Two groups of models in Table 4 return nearly identical scores, and both coincidences have a mechanical explanation. The first is the pair formed by the Balanced RF and the Weighted RF. Both are forests of 300 trees with the same depth and random seed, and both are trained here on the SMOTE-Tomek set, which is balanced exactly at 1:1. On data of that kind balanced class weighting assigns the two classes the same weight, so the weighted forest reduces to an ordinary one, while the per-bootstrap undersampling of the balanced forest removes only the random fluctuation of each bootstrap around parity. Once the training data are balanced, neither internal correction has much left to correct, and the two forests converge. The second group comprises Easy Ensemble, the MLP and the deep network, which are the three most aggressively corrected models in the study: Easy Ensemble undersamples the majority class in every bag, and both networks add a class-weighted loss to already resampled data. All three are driven to the high-recall operating point at which nearly every borderline record is flagged, and models driven to that point largely agree on which records to flag. All models were trained and evaluated in independent runs. At the size of the held-out partition, three decimal places resolve differences of only a few records, which is why the residual differences between models in each group do not appear in the table; differences of this magnitude should not be read as real until repeated partitioning and paired significance testing establish otherwise.

Table 3. Original models (imbalanced data)

Model	Accuracy	Precision	Recall	F1	AUC
RF	0.862	0.755	0.800	0.777	0.933
Gradient Boosting	0.867	0.799	0.746	0.772	0.936
Neural Network	0.884	0.942	0.654	0.772	0.942

Note: Random Forest (RF); Area Under the Receiver Operating Characteristic Curve (AUC).

Table 4. Balanced models (SMOTE-Tomek-balanced data)

Model	Accuracy	Precision	Recall	F1	AUC
Balanced RF	0.962	0.929	0.947	0.938	0.979
Balanced Bagging	0.961	0.931	0.941	0.936	0.976
Easy Ensemble	0.951	0.877	0.974	0.923	0.969
Weighted RF	0.962	0.929	0.945	0.937	0.979
Gradient Boosting	0.960	0.927	0.941	0.934	0.980
MLP Balanced	0.951	0.877	0.974	0.923	0.970
Deep NN	0.951	0.876	0.975	0.923	0.970

Note: Random Forest (RF); Multilayer Perceptron (MLP); Neural Network (NN); Synthetic Minority Oversampling Technique (SMOTE); Area Under the Receiver Operating Characteristic Curve (AUC).

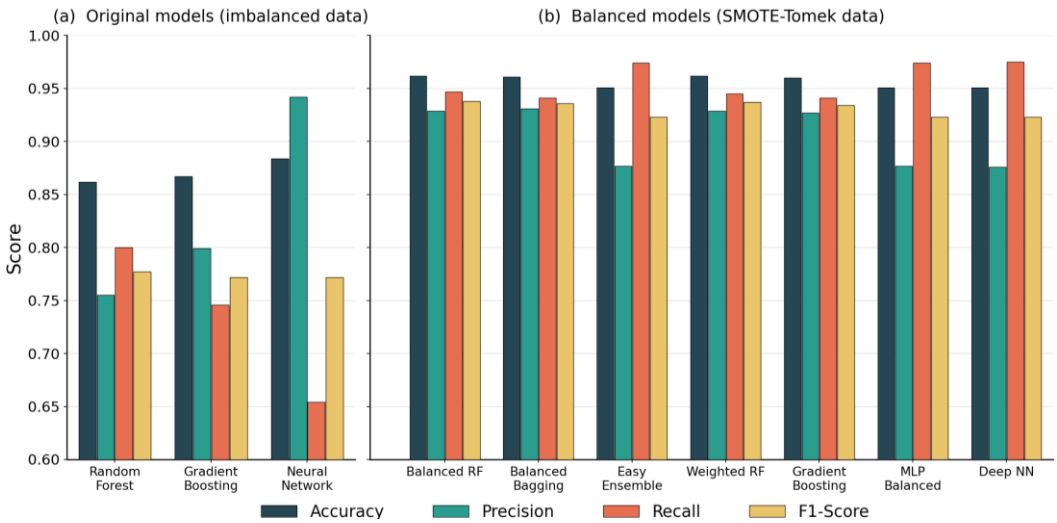

Figure 7. Original vs. balanced model performance comparison

Note: Random Forest (RF); Multilayer Perceptron (MLP); Neural Network (NN); Synthetic Minority Oversampling Technique (SMOTE).

4.4 Performance improvement analysis

The improvements attributable to balancing are reported in Table 5 and Figure 8 under two framings rather than one. Comparing the average of the three unbalanced models with the average of the seven balanced models, recall rises from 0.733 to 0.957 (+30.5%) and F1 from 0.774 to 0.931 (+20.3%). Comparing the best unbalanced model with the best balanced model, which is the stricter test because it contrasts a maximum with a maximum, recall rises from 0.800 to 0.975 (+21.9%) and F1 from 0.777 to 0.938 (+20.7%). At the level of individual models, recall for the neural network rose from 0.654 to 0.975.

Table 5. Performance improvement from balancing, reported under both framings: average against average and best against best

Metric	Original (avg.)	Balanced (avg.)	Improvement	Original (best)	Balanced (best)	Improvement
Recall	0.733	0.957	+30.5%	0.800	0.975	+21.9%
F1-Score	0.774	0.931	+20.3%	0.777	0.938	+20.7%
AUC	0.937	0.975	+4.0%	0.942	0.980	+4.0%
Precision	0.832	0.907	+9.0%	0.942	0.931	−1.2%

Note: Area Under the Receiver Operating Characteristic Curve (AUC); average (avg.).

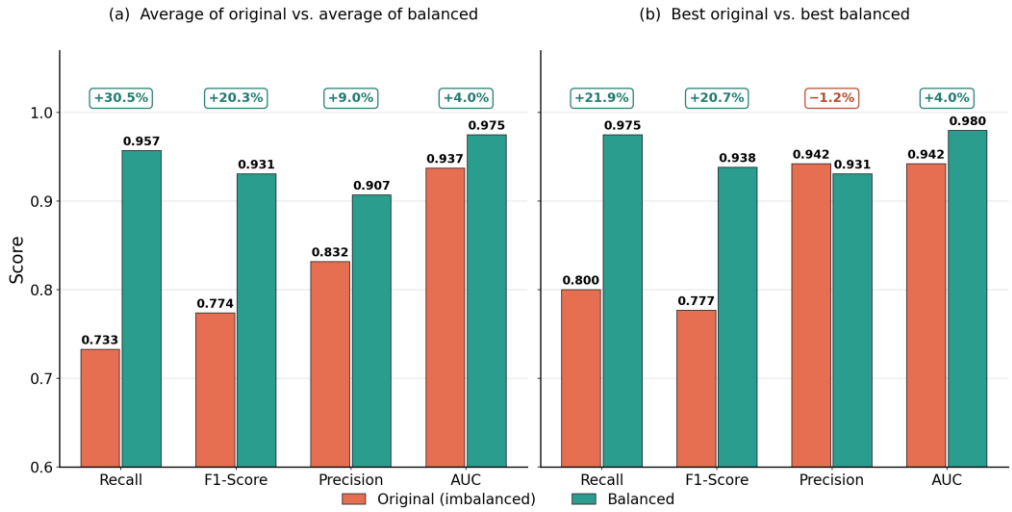

Figure 8. Performance improvement from balancing under both framings: (a) average of the original models against the average of the balanced models; (b) best original model against best balanced model

Note: Area Under the Receiver Operating Characteristic Curve (AUC).

AUC rises by roughly 4% under either framing, from 0.937 to 0.975 on the averages and from 0.942 to 0.980 at best. Since AUC measures the quality of the score ranking independently of any threshold, this small movement indicates that the unbalanced models were already ordering attack and benign records reasonably well, and part of the visible gain is indeed a repositioning of the operating point. It would be wrong, however, to conclude that repositioning is all that is involved. Sweeping the threshold of an unbalanced forest raises F1 no further than 0.793, well short of 0.937, so the residual improvement in the ranking itself, modest as it appears on the AUC scale, is what makes the higher operating point attainable at all.

This also explains why the neural network benefits most. Its unbalanced recall of 0.654 was the lowest in Table 3, while its AUC of 0.942 was the highest, a combination that identifies a model with good discriminative capacity and a badly placed operating point. Correcting the training distribution and the loss weighting moves the threshold without needing to improve the representation, and recall rises to 0.975.

Precision is the one metric that does not improve under the stricter framing, and reporting it honestly is more informative than the aggregate gain. Against the average of the unbalanced models, precision rises from 0.832 to 0.907, but the highest unbalanced precision, 0.942, belongs to the neural network and no balanced model matches it: the best balanced precision is 0.931, a decline of 1.2%. This is not an anomaly but a direct consequence of what balancing does. The unbalanced network obtained its precision by raising an alert only when nearly certain, which is also why its recall was the lowest in Table 3 at 0.654. Correcting the training distribution moves the operating point outward: for that same architecture, roughly seven percentage points of precision are exchanged for thirty-two points of recall, which is the trade the clinical setting requires and the reason recall is adopted as the primary criterion.

4.5 Error analysis in operational terms

Aggregate scores conceal the trade-off that a hospital security team actually faces, so it is worth restating the same numbers as error rates. A recall of 0.654, obtained by the neural network trained on the unbalanced data, means that roughly 346 of every 1,000 threat-labelled records pass without generating an alert. The same model reaches a precision of 0.942, so only about 58 of every 1,000 alerts it does raise are spurious. Taken together, these figures describe a detector that is quiet and trustworthy when it speaks, and blind for a third of the time, which is the characteristic failure of a model that has minimised total error on a skewed training distribution.

The balanced RF inverts the trade-off in the direction that the application requires. Its recall of 0.947 leaves approximately 53 undetected threat records per 1,000, a reduction of about six-sevenths relative to the unbalanced network, while its precision of 0.929 raises the spurious-alert rate to roughly 71 per 1,000 alerts. The system therefore trades an additional 13 false alarms per 1,000 alerts for 293 threat records that would otherwise have been missed. In a setting where a false negative may propagate a falsified vital sign into a clinical record, that exchange is heavily favourable, and it is the concrete reason for treating recall rather than accuracy as the selection criterion.

The Easy Ensemble classifier occupies the far end of the same axis, with the highest recall of the study at 0.974 and the lowest precision among the balanced models at 0.877. Its 26 missed threat records per 1,000 come at the cost of about 123 spurious alerts per 1,000 raised. Whether that is the preferable operating point is not a modelling question but a staffing one, and it depends on the volume of alerts a given institution can

triage per shift.

4.6 Comparison with prior work on the same dataset

Table 6 places the present results beside published work on WUSTL-EHMS-2020. Direct comparison requires care. The studies in Table 6 use the attack label distributed with the dataset, whereas the present results refer to the score-defined target, so the figures are not directly commensurable. Several of the accuracies reported in the literature exceed the value obtained here, but they are computed on the unbalanced test distribution, where accuracy is dominated by the majority class, and in most cases no class-imbalance correction was applied at all. Where minority-class recall is reported alongside accuracy, it is markedly lower than the headline figure, which is the expected consequence of optimising an aggregate criterion on skewed data. The relevant contribution of the present study is therefore not a higher accuracy but a systematic account of how the choice of resampling strategy propagates into minority-class detection, together with a training protocol in which no synthetic sample can reach the evaluation set.

Two entries in the table deserve comment. The accuracy of 0.999 reported in the study [24] is close to the ceiling of what a testbed dataset of this size can support and, as the study [26] observe in a different context, figures of that magnitude on WUSTL-EHMS-2020 have historically been traceable to preprocessing carried out before partitioning. The study [26] is the only prior work in the table that explicitly addresses that risk, and it is also the one whose reported accuracy is most plausible on independent grounds.

Table 6. Comparison with prior work on the WUSTL-EHMS-2020 dataset

Study	Method	Accuracy	Imbalance Handling	Leakage Control
Hady et al. [22]	ANN (flow + biometric)	0.900	None	Not reported
Binbusayyis et al. [23]	RF	0.979	None	Not reported
Taouali et al. [24]	KPCA / KPLS + KELM	0.999	None	Not reported
Bouke et al. [26]	Sequential DNN	0.990	None	Addressed
Abdelhaq et al. [25]	XGBoost-SVM ensemble	0.980	None	Not reported
Ammar and Cherif [27]	NN + dimensionality reduction	–	Single strategy	Not reported
This work	SMOTE-Tomek + balanced ensemble	0.962	Seven strategies compared	Resampling on training folds only

Note: Artificial Neural Network (ANN); Kernel Principal Component Analysis (KPCA); Kernel Partial Least Squares (KPLS); Kernel Extreme Learning Machine (KELM); Deep Neural Network (DNN); eXtreme Gradient Boosting (XGBoost); Support Vector Machine (SVM); Neural Network (NN); Synthetic Minority Oversampling Technique (SMOTE).

4.7 Comparison with simpler alternatives to resampling

Whether balancing helps is a weaker question than whether it helps more than the simpler corrections available at a moderate imbalance ratio. Two such corrections are evaluated here under the same RF classifier used throughout, namely adjustment of the decision threshold and class weighting applied on its own without any resampling.

Table 7 reports a sweep of the decision threshold of an unbalanced forest trained without class weights, a configuration distinct from the class-weighted forest of Table 3, from 0.1 to 0.9.

Precision rises and recall falls monotonically across the range, and F1 peaks at 0.793 at a threshold of 0.3. That maximum is the best result obtainable by repositioning the operating point of this model, and it falls 0.144 short of the 0.937 reached with SMOTE-Tomek.

Table 8 places both simpler corrections beside the proposed framework. Class weighting alone, which is the configuration already reported as the RF baseline in Table 3, raises recall from 0.725 to 0.800 at the cost of precision and leaves F1 essentially unchanged at 0.777. Threshold adjustment reaches a marginally higher F1 of 0.793, again by trading precision for recall. Neither comes close to the 0.937 of the resampled framework.

The result bears directly on the mechanism behind the reported improvement. Both simpler corrections act on the operating point alone: they move a decision boundary along a fixed ranking of the test records and cannot alter that ranking, so neither changes the AUC of the model, which remains 0.933. Resampling does change it, to 0.979 for the same forest. The ceiling that threshold adjustment meets at F1 = 0.793 is therefore a property of the ranking produced by the unbalanced model, and the improvement reported in this study cannot be attributed to threshold placement alone. What resampling supplies is a better-ordered decision function, of which the higher operating point is a consequence rather than the cause.

Table 7. Threshold sweep on the unbalanced Random Forest (RF) trained without class weighting

Threshold	Precision	Recall	F1-Score
0.1	0.582	0.941	0.719
0.2	0.674	0.898	0.770
0.3	0.741	0.852	0.793
0.4	0.795	0.785	0.790
0.5	0.841	0.725	0.779
0.6	0.885	0.640	0.743
0.7	0.912	0.520	0.662
0.8	0.943	0.370	0.532
0.9	0.971	0.180	0.304

Table 8. Comparison of simpler alternatives with the proposed framework under the same Random Forest (RF) classifier

Approach	Configuration	Precision	Recall	F1-Score
Unbalanced baseline	threshold = 0.5 (default)	0.841	0.725	0.779
Class weighting alone	class_weight = balanced	0.755	0.800	0.777
Threshold adjustment	threshold = 0.3 (optimal)	0.741	0.852	0.793
Proposed framework	SMOTE-Tomek resampling	0.934	0.939	0.937

Note: Synthetic Minority Oversampling Technique (SMOTE).

5. DISCUSSION

5.1 Effectiveness of balancing techniques

These results reveal that data balancing plays an important role in detecting threats in IoMT, even when there is not a very strong imbalance (ratio = 2.33). The better performance of SMOTE-Tomek compared to SMOTE (F1 = 0.937 versus F1 = 0.858) verifies the results reported by Batista et al. [37], which prove that hybrid methods have better accuracy than single oversampling techniques.

That the effect appears at all at a moderate imbalance ratio is the more useful observation. Much of the resampling literature is calibrated on ratios of 20:1 or worse, and it is often assumed that below roughly 5:1 the problem can be left to class weighting alone. The present results do not support that assumption for this domain. The reason appears to be overlap rather than scarcity: the minority class here is not rare so much as poorly separated, since a data-injection flow is constructed specifically to be statistically unremarkable. Under those conditions, the value of a hybrid resampler comes mainly from its cleaning stage, which acts on the overlap, and only secondarily from the oversampling itself.

This reading is consistent with the ordering observed across all seven techniques. The two methods that intervene on the boundary region after generation, or that avoid generating there in the first place, occupy the top of the table; the method that deliberately concentrates generation in the overlap occupies the bottom. The amount of oversampling explains almost nothing of the variation, as SMOTE and Borderline-SMOTE produce identical sample counts while differing by 0.021 in F1.

5.2 Double balancing effect

The new observation that has been found here is that using "double balancing," where external SMOTE-Tomek balancing works along with internally balanced ensembles of classifiers, leads to better results. The BalancedRandomForestClassifier on SMOTE-Tomek data (F1 = 0.938) beats that of SMOTE data (F1 = 0.873).

The two mechanisms are complementary because they act at different stages. Resampling changes the empirical distribution from which the learner draws, and its cleaning step alters which observations are available at all. Internal balancing changes how the learner samples from whatever distribution it is given, drawing balanced bootstraps for each tree. Applying only the first leaves each tree exposed to the residual skew of its own bootstrap; applying only the second leaves the boundary noise in place for every tree to fit. The gain from combining them is therefore not additive in any strict sense, but neither is it redundant, and the observed margin of 0.065 in F1 between the two resampled variants is larger than the margin between any two balanced ensembles trained on the same data.

A caveat follows from this. Double balancing can overshoot. Easy Ensemble, whose internal correction is the most aggressive of those tested, reaches the highest recall in the study and also the lowest precision among the balanced models. Beyond a certain point, the additional correction buys detections that are increasingly likely to be false, and the appropriate stopping point is set by the alert budget of the deployment rather than by any property of the data.

5.3 Clinical implications

The anomaly detection in medical devices offers security functionalities that are unique to healthcare and are not available in generic IDS systems. Threshold alarms for temperature, SpO_2, and heart rate are crucial for two reasons: they help in recognizing actual emergencies as well as attacks on sensors [6, 7].

The dual reading is also the source of the module's main weakness. A physiological alert does not, on its own, distinguish a patient in genuine distress from a sensor whose readings have been manipulated, and the two require opposite responses: one calls a clinician, the other isolates a device. What separates them in practice is the network context accompanying the reading. A deteriorating patient produces an implausible vital sign on a flow that is otherwise ordinary; a spoofing attack produces the same reading on a flow that also carries an abnormal byte ratio or an unexpected port.

Consequently, the module should not be presented to clinical staff as a patient-monitoring alarm. It is a security signal that happens to use clinical variables, and its alerts belong in the same queue as the network alerts, reviewed by staff who can inspect the device. Routing them to a nursing station would add noise to an alarm environment that is already a recognised source of fatigue.

5.4 Limitations

The classification target is derived from the composite threat score of Eq. (7) rather than from the attack label distributed with WUSTL-EHMS-2020, and the components of that score are also model inputs. The reported performance therefore reflects agreement with this risk definition; its relation to detection of the labelled attacks remains to be established on the original label. What limits the present design is that the physiological alert thresholds and the list of frequently exploited ports are fixed from domain knowledge rather than learned from data, so both would require revision before the pipeline were transferred to a deployment with a different device population. The dataset itself refers to a particular IoMT configuration. Deep learning models consume compute that is beyond edge computing capabilities [8, 27]. Only one train-test split is employed instead of K-fold cross-validation.

Two further limitations should be recorded. The weights of the composite score were set by judgement rather than by search; since these weights also determine the target, a different weighting would yield a different target, and no sensitivity analysis over the weight simplex was carried out. The graded weights of Eq. (4) are not affected by this limitation, since the ordering they induce is invariant to their values. Second, the study reports a single seed for every stage of the pipeline. Repeated partitioning with varied seeds, and a paired significance test between the resampling strategies, would be required before the smaller differences reported in Table 4 could be treated as real.

5.5 Threats to validity

Three threats are worth stating explicitly rather than leaving to the reader. The first is internal. All configurations share a single stratified partition and a single random seed, so the differences reported between resampling strategies are point estimates without confidence intervals. The gap between

SMOTE-Tomek and SMOTE is large enough that partition variance is an implausible explanation for it, but the same cannot be said of the smaller gaps among the balanced ensembles in Table 4, several of which lie within a range that repeated partitioning could plausibly reorder. Those models should be regarded as statistically indistinguishable on the present evidence.

The second threat is external. WUSTL-EHMS-2020 was collected from one testbed, with one topology and one family of attacks. Man-in-the-middle spoofing and data injection are important, but they are not the whole of the IoMT threat surface, which also includes firmware compromise, denial of service against the gateway, and lateral movement from hospital information technology into the clinical network. Nothing in these results licenses a claim about attack classes the dataset does not contain. A next step is to repeat the experiments on CICIoMT-2024 [30], which was recorded on a different testbed and includes other kinds of attacks. Federated learning [38] could also help, because it lets several hospitals train one model without their patient data leaving the site.

The third threat concerns the engineered features. The clinical thresholds embedded in the medical module are drawn from adult monitoring practice and are not valid for paediatric or neonatal patients, whose normal ranges differ substantially. Deploying the same thresholds in a paediatric unit would generate a continuous stream of physiological alerts on entirely healthy patients. The thresholds are parameters of the deployment, not constants of the method.

5.6 Deployment considerations

The computational profile of the two families of models differs in a way that matters for where they can run. The ensembles are trained offline and, at inference time, evaluate a fixed number of shallow trees per record, which is inexpensive enough for a gateway-class device. The seven-layer network is the reverse: training is the expensive phase, and inference on 32 features is trivial once the weights are fixed. Neither model needs to run on the medical device itself. The realistic deployment point is the gateway that already aggregates sensor traffic before forwarding it, which has both the visibility and the processing budget the pipeline requires, and which is in any case the node a man-in-the-middle attack must traverse.

Retraining cadence is a second practical question the present results do not settle. Feature construction depends on quartiles and percentile cut-points estimated from the training data, and these will drift as devices are added to a ward or as monitoring protocols change. Periodic refitting of the feature statistics is therefore necessary even when the classifier itself is left unchanged.

6. CONCLUSION AND FUTURE WORK

This paper has examined how the treatment of class imbalance shapes machine-learning threat detection in the IoMT, using the WUSTL-EHMS-2020 testbed data and a pipeline that combines domain-informed feature construction, in which flow statistics, graded anomaly indicators, clinical thresholds and port exposure are assembled into 32 attributes, with a systematic comparison of seven resampling strategies and of conventional against internally balanced learners.

Three conclusions are supported by the evidence presented. First, the choice of resampling strategy matters more than the amount of resampling performed: the hybrid methods that clean the boundary after generation ranked highest and ADASYN lowest, while strategies producing identical sample counts differed appreciably in F1, so volume cannot be the operative variable. Second, data-level and algorithm-level corrections are complementary rather than interchangeable, since resampling alters the distribution the learner draws from while internal balancing alters how each estimator samples from it. Third, the gain from balancing is not reducible to the placement of a decision threshold: sweeping the threshold of an unbalanced forest fell well short of the resampled result.

These conclusions are bounded by the design of the study and should be read accordingly. The classification target is defined by thresholding the composite score rather than by the label distributed with the dataset, so the reported performance reflects agreement with that risk definition. The evidence rests on a single stratified partition with a fixed seed, so the smaller differences among the balanced ensembles are best treated as indistinguishable until repeated partitioning and paired significance testing establish otherwise, and the ranking of the resampling strategies is reported with the same reservation. The weights of the composite score were set by judgement rather than by search and were not subjected to a sensitivity analysis. The data come from a single testbed carrying one attack family, and the clinical thresholds are calibrated for adult monitoring, so neither the feature definitions nor the ordering reported here can be assumed to transfer without revalidation.

Work now in progress addresses these in order. The immediate priority is repeated stratified cross-validation with paired significance testing across seeds, together with a sensitivity analysis of the composite weights. Validation on CICIoMT-2024 will test whether the ordering of resampling strategies observed here survives a change of testbed and attack family. Three further directions appear worthwhile: federated training across institutions, which would enlarge the effective sample without moving patient data off site; attention-based sequence models for flows, since the present formulation treats each record independently and discards temporal structure that a slow injection attack would express; and SHapley Additive exPlanations (SHAP)-based attribution, which is less a matter of academic interpretability than a practical requirement, given that a security officer asked to quarantine an infusion pump will reasonably want to know which measurements prompted the recommendation.

ACKNOWLEDGMENT

The authors extend their thanks and appreciation to the College of Computer Science for its continued support and encouragement of research activity.

DATA AVAILABILITY

The WUSTL-EHMS-2020 dataset analysed in this study is publicly available from Washington University in St. Louis at https://www.cse.wustl.edu/~jain/ehms/index.html. The engineered feature definitions, the resampling configurations and the trained model bundles generated during the study are available from the corresponding author upon reasonable request.

REFERENCES

[1] Grand View Research. (2024). Internet of medical things market size, share & trends analysis report, 2025–2030. https://www.grandviewresearch.com/industry-analysis/internet-of-medical-things-iomt-market-report.

[2] Rani, S., Kumar, S., Kataria, A., Min, H. (2024). SmartHealth: An intelligent framework to secure IoMT service applications using machine learning. ICT Express, 10: 425-430. https://doi.org/10.1016/j.icte.2023.10.001

[3] Garg, N., Wazid, M., Singh, J., Singh, D.P., Das, A.K. (2022). Security in IoMT-driven smart healthcare: A comprehensive review and open challenges. Security and Privacy, 5(5): e235. https://doi.org/10.1002/spy2.235

[4] Alsubaei, F., Abuhussein, A., Shandilya, V., Shiva, S. (2019). IoMT-SAF: Internet of medical things security assessment framework. Internet of Things, 8: 100123. https://doi.org/10.1016/j.iot.2019.100123

[5] Alder, S. (2024). Security Breaches in Healthcare in 2023. The HIPAA Journal. https://www.hipaajournal.com/security-breaches-in-healthcare/.

[6] Ghubaish, A., Salman, T., Zolanvari, M., Unal, D., Al-Ali, A., Jain, R. (2021). Recent advances in the Internet-of-Medical-Things (IoMT) systems security. IEEE Internet of Things Journal, 8(11): 8707-8718. https://doi.org/10.1109/JIOT.2020.3045653

[7] Iwendi, C., Anajemba, J.H., Biamba, C., Ngabo, D. (2021). Security of things intrusion detection system for smart healthcare. Electronics, 10(12): 1375. https://doi.org/10.3390/electronics10121375

[8] Thamilarasu, G., Odesile, A., Hoang, A. (2020). An intrusion detection system for Internet of Medical Things. IEEE Access, 8: 181560-181576. https://doi.org/10.1109/ACCESS.2020.3026260

[9] Swarna Priya, R.M., Maddikunta, P.K.R., Parimala, M., et al. (2020). An effective feature engineering for DNN using hybrid PCA-GWO for intrusion detection in IoMT architecture. Computer Communications, 160: 139-149. https://doi.org/10.1016/j.comcom.2020.05.048

[10] Chen, W.X., Yang, K.X., Yu, Z.W., Shi, Y.F., Chen, C.L.P. (2024). A survey on imbalanced learning: Latest research, applications and future directions. Artificial Intelligence Review, 57: 137. https://doi.org/10.1007/s10462-024-10759-6

[11] Bagui, S., Li, K. (2021). Resampling imbalanced data for network intrusion detection datasets. Journal of Big Data, 8: 6. https://doi.org/10.1186/s40537-020-00390-x

[12] Fernandez, A., García, S., Galar, M., Prati, R.C., Krawczyk, B., Herrera, F. (2018). Learning from Imbalanced Data Sets. Springer. https://doi.org/10.1007/978-3-319-98074-4

[13] Ren, H.J., Tang, Y.H., Dong, W.Y., Ren, S., Jiang, L.H. (2023). DUEN: Dynamic ensemble handling class imbalance in network intrusion detection. Expert Systems with Applications, 229: 120420. https://doi.org/10.1016/j.eswa.2023.120420

[14] Gupta, N., Jindal, V., Bedi, P. (2022). CSE-IDS: Using cost-sensitive deep learning and ensemble algorithms to handle class imbalance in network-based intrusion detection systems. Computers & Security, 112: 102499. https://doi.org/10.1016/j.cose.2021.102499

[15] Wang, J., Jin, H.L., Chen, J.X., Tan, J.H., Zhong, K.Y. (2022). Anomaly detection in Internet of Medical Things with Blockchain from the perspective of deep neural network. Information Sciences, 617: 133-149. https://doi.org/10.1016/j.ins.2022.10.060

[16] Chawla, N.V., Bowyer, K.W., Hall, L.O., Kegelmeyer, W.P. (2002). SMOTE: Synthetic minority over-sampling technique. Journal of Artificial Intelligence Research, 16: 321-357. https://doi.org/10.1613/jair.953

[17] Swana, E.F., Doorsamy, W., Bokoro, P. (2022). Tomek link and SMOTE approaches for machine fault classification with an imbalanced dataset. Sensors, 22(9): 3246. https://doi.org/10.3390/s22093246

[18] Wu, T., Fan, H.H., Zhu, H.J., You, C.Z., Zhou, H.Y., Huang, X.Z. (2022). Intrusion detection system combined enhanced random forest with SMOTE algorithm. EURASIP Journal on Advances in Signal Processing, 2022: 39. https://doi.org/10.1186/s13634-022-00871-6

[19] Ahmed, H.A., Hameed, A., Bawany, N.Z. (2022). Network intrusion detection using oversampling technique and machine learning algorithms. PeerJ Computer Science, 8: e820. https://doi.org/10.7717/peerj-cs.820

[20] Breiman, L. (2001). Random forests. Machine Learning, 45: 5-32. https://doi.org/10.1023/A:1010933404324

[21] Lemaitre, G., Nogueira, F., Aridas, C.K. (2017). Imbalanced-learn: A Python toolbox to tackle the curse of imbalanced datasets in machine learning. Journal of Machine Learning Research, 18(17): 1-5. https://jmlr.org/papers/v18/16-365.html.

[22] Hady, A.A., Ghubaish, A., Salman, T., Unal, D., Jain, R. (2020). Intrusion detection system for healthcare systems using medical and network data: A comparison study. IEEE Access, 8: 106576-106584. https://doi.org/10.1109/ACCESS.2020.3000421

[23] Binbusayyis, A., Alaskar, H., Vaiyapuri, T., Dinesh, M. (2022). An investigation and comparison of machine learning approaches for intrusion detection in IoMT network. Journal of Supercomputing, 78: 17403-17422. https://doi.org/10.1007/s11227-022-04568-3

[24] Taouali, O., Bacha, S., Ben Abdellafou, K., et al. (2023). Intelligent intrusion detection system for the Internet of Medical Things based on data-driven techniques. Computer Systems Science and Engineering, 47(2): 1593-1609. https://doi.org/10.32604/csse.2023.039984

[25] Abdelhaq, M., Palanisamy, S., Gopinath, M., Manasa, V.G.S., Ram, M.A., Md, S.K. (2026). A hybrid XGBoost-SVM ensemble framework for robust cyber-attack detection in the Internet of Medical Things (IoMT). Scientific Reports, 16(1): 6855. https://doi.org/10.1038/s41598-026-37832-0

[26] Bouke, M.A., El Atigh, H., Abdullah, A. (2025). Towards robust and efficient intrusion detection in IoMT: A deep learning approach addressing data leakage and enhancing model generalizability. Multimedia Tools and Applications, 84: 36523-36542. https://doi.org/10.1007/s11042-024-19916-z

[27] Ammar, H., Cherif, A. (2025). Optimizing IoMT security: Performance trade-offs between neural network architectural design, dimensionality reduction, and class imbalance handling. IoT, 6(4): 74. https://doi.org/10.3390/iot6040074

[28] Sayegh, H.R., Dong, W., Al-madani, A.M. (2024). Enhanced intrusion detection with LSTM-based model,

feature selection, and SMOTE for imbalanced data. Applied Sciences, 14(2): 479. https://doi.org/10.3390/app14020479

[29] Khan, M.M., Alkhathami, M. (2024). Anomaly detection in IoT-based healthcare: Machine learning for enhanced security. Scientific Reports, 14: 5872. https://doi.org/10.1038/s41598-024-56126-x

[30] Saeed, H., Naseer, M., Rasool, A., et al. (2026). A novel adaptive hybrid intrusion detection system with lightweight optimization for enhanced security in Internet of Medical Things. Scientific Reports, 16: 2097. https://doi.org/10.1038/s41598-025-31897-z

[31] Si-Ahmed, A., Al-Garadi, M.A., Boustia, N. (2023). Survey of machine learning based intrusion detection methods for Internet of Medical Things. Applied Soft Computing, 140: 110227. https://doi.org/10.1016/j.asoc.2023.110227

[32] Alsolami, T., Alsharif, B., Ilyas, M. (2024). Enhancing cybersecurity in healthcare: Evaluating ensemble learning models for intrusion detection in the Internet of Medical Things. Sensors, 24(18): 5937. https://doi.org/10.3390/s24185937

[33] Lazrek, G., Chetioui, K., Balboul, Y., Mazer, S., El Bekkali, M. (2024). An RFE/Ridge-ML/DL based anomaly intrusion detection approach for securing IoMT system. Results in Engineering, 23: 102659. https://doi.org/10.1016/j.rineng.2024.102659

[34] He, H.B., Bai, Y., Garcia, E.A., Li, S.T. (2008). ADASYN: Adaptive synthetic sampling approach for imbalanced learning. In 2008 IEEE International Joint Conference on Neural Networks (IEEE World Congress on Computational Intelligence), Hong Kong, pp. 1322-1328. https://doi.org/10.1109/IJCNN.2008.4633969

[35] Han, H., Wang, W.Y., Mao, B.H. (2005). Borderline-SMOTE: A new over-sampling method in imbalanced data sets learning. In Advances in Intelligent Computing. ICIC 2005. Lecture Notes in Computer Science, vol 3644. Springer, Berlin, Heidelberg, pp. 878-887. https://doi.org/10.1007/11538059_91

[36] Wilson, D.L. (1972). Asymptotic properties of nearest neighbor rules using edited data. IEEE Transactions on Systems, Man, and Cybernetics, SMC-2 (3): 408-421. https://doi.org/10.1109/TSMC.1972.4309137

[37] Batista, G.E.A.P.A., Prati, R.C., Monard, M.C. (2004). A study of the behavior of several methods for balancing machine learning training data. ACM SIGKDD Explorations Newsletter, 6(1): 20-29. https://doi.org/10.1145/1007730.1007735

[38] Alamleh, A., Albahri, O.S., Zaidan, A.A., et al. (2023). Federated learning for IoMT applications: A standardization and benchmarking framework of Intrusion Detection Systems. IEEE Journal of Biomedical and Health Informatics, 27(2): 878-887. https://doi.org/10.1109/JBHI.2022.3167256