The Effect of Increased Dimensionality on Detecting Malicious IoMT Network Traffic*

Jayden Alonzo-Estrada, Anthony Mora, Abraham Avila and Ram Basnet†

Colorado Mesa University, Grand Junction, U.S.
{jjalonzo-estra, ajmora}@mavs.coloradomesa.edu,
avila.abraham117@gmail.com, rbasnet@coloradomesa.edu

Abstract

This paper explores the impact of feature dimensionality on the performance of machine learning (ML) models in detecting malicious traffic within Internet of Medical Things (IoMT) Wi-Fi networks. Using the CICIoMT2024 dataset, we compare two feature extraction techniques—DPKT, which yields 39 packet-level features, and CICFlowMeter, which produces 79 flow-level features across six models: XGBoost, AdaBoost, Random Forest, Multilayer Perceptron (MLP), Logistic Regression, and Decision Tree. Models were evaluated under binary and multi-class classification tasks. Our results show that Logistic Regression and MLP significantly benefit from higher-dimensional feature sets, with MLP's F1 score increasing from 0.109 to 0.498 in multi-class classification. In contrast, ensemble models such as XGBoost and Random Forest achieve high baseline performance even with low-dimensional inputs, with marginal gains, and sometimes losses, from additional features. However, richer features also introduce substantial computational overhead. For instance, AdaBoost required over 25,000 minutes (~17 days) to train on high-dimensional data, making it unsuitable for real-time scenarios.

Keywords: IoMT, ML, IoT, CICIoMT2024, CICFlowMeter, DPKT, IPS, IDS.

1 Introduction

The IoMT is reshaping modern healthcare by enabling real-time monitoring, remote diagnostics, and automated treatment through interconnected devices such as smart infusion pumps, wearable health trackers, and wireless vital sign monitors (Dadkhah et al., 2024; O'Brien et al., 2018). While these technologies enhance patient outcomes and expand access

* Proceedings of the 9th International Conference on Mobile Internet Security (MobiSec'25), Article No. 19, December 16-18, 2025, Sapporo, Japan. © The copyright of this paper remains with the author(s).

† Corresponding author

to care, they also introduce significant cybersecurity risks. Many IoMT devices lack robust embedded security due to strict resource constraints and real-time operational requirements, making them especially vulnerable to attack vectors uncommon in traditional IT infrastructure (Khaled, 2022; Rehman et al., 2025).

These limitations create an attractive surface for adversaries looking to compromise medical networks. Protocols commonly used in IoMT environments—such as Message Queuing Telemetry Transport (MQTT) and Bluetooth—are particularly susceptible to threats like spoofing, man-in-the-middle (MITM) attacks, and Denial-of-Service (DoS) campaigns (Dadkhah et al., 2024). The CICIoMT2024 dataset reflects these concerns by simulating real-world attacks across a hybrid testbed of virtual and physical IoMT devices, capturing sophisticated multi-vector threats over Wi-Fi and MQTT channels (Dadkhah et al., 2024). Complementary research, such as the HybridIoMT framework (Taha et al., 2025), also highlights protocol-level weaknesses and proposes dual-phase ML filters as a mitigation strategy.

ML-powered IDS and IPS have become increasingly popular due to their ability to learn complex attack patterns and adapt to diverse network environments. However, the performance of these systems is closely tied to the quality and dimensionality of input features (Dadkhah et al., 2024; Rehman et al., 2025). Datasets with limited feature representation often fail to generalize, while high-dimensional data impose steep computational demands, especially problematic for deployment on edge-based or embedded IoMT platforms (Agus, 2025; Rehman et al., 2025; Zachos et al., 2025).

In this work, we investigate the role of feature dimensionality in the performance of six ML models: XGBoost, AdaBoost, Random Forest, MLP, AdaBoost, and Logistic Regression. We compare two feature extraction techniques, DPKT, which produces 39 low-level packet features, and CICFlowMeter, yielding 79 higher-level flow features on the CICIoMT2024 dataset. Our goal is to quantify the effect of dimensionality on model generalization, training time, and suitability for binary, class, and sub-class classification tasks in constrained IoMT environments.

This research project evaluates anomaly detection systems across a custom-built IoMT testbed with 15 simulated and 7 real devices. Some, but not all, of the simulated devices in the testbed include a Fall detector, infusion pump, and iHealth Smart Wireless Glucose-Monitoring system. The testbed also incorporates real IoMT devices such as the Sense U Baby Monitor, Blink Mini, and the Ecobee Camera (Dadkhah et al., 2024). These devices provide a detailed taxonomy of attacks, including ARP Spoofing, MQTT Malformed Data, Port Scans, SYN Floods, and other DoS/DDoS variants. It confirms the relevance and realism of these attacks in datasets like CICIoMT2024 and supports their use in ML-based IDS evaluations.

2 Related Works

Several prior studies have investigated the impact of feature extraction on IDS, particularly within IoT and IoMT contexts.

One foundational work by Mohanad Sarhan et al. (Sarhan et al., 2024). explored how different feature extraction techniques affect ML model performance on network intrusion detection system (NIDS) datasets. They applied dimensionality reduction methods such as Principal Component Analysis (PCA), Linear Discriminant Analysis (LDA), and Auto-Encoders (AE) across three benchmark NIDS datasets, training six distinct classifiers, including Deep Feedforward Networks, Convolutional Neural Networks (CNNs), Residual Neural Networks (RNNs), Decision Trees, Logistic Regression, and Naive Bayes. The study concluded that no universal combination of features and models generalized well across all datasets. Notably, classification accuracy plateaued beyond ten features, and the authors recommended a standardized feature set for broader applicability.

Another relevant study by Ola Salman et al. (Salman et al., 2019) proposed an ML-based framework for abnormal traffic detection at the network edge. Their approach emphasizes resource efficiency by extracting only primitive packet features from small flow segments (e.g., 16 packets) to minimize memory and processing overhead. The models evaluated included Decision Trees, Random Forests, and several deep learning architectures such as CNNs and RNNs. Their findings suggest that basic statistical features, without deep packet inspection, are sufficient to distinguish traffic types with high accuracy, making the approach practical for real-time, edge-based deployments.

In a study, directly leveraging the CICIoMT2024 dataset by Mohamadi, Alireza et al. (Mohammadi et al., 2024) developed a CNN-based model for multi-class attack detection in IoMT networks. Their rationale lies in CNNs' strength in modeling temporal and high-dimensional data. Their model was benchmarked against AdaBoost, Random Forest, MLP, and Logistic Regression, outperforming them across all classification tasks. Importantly, the authors noted that the CICIoMT2024 dataset has several closely related attack sub-classes, which pose challenges for traditional ML models. They advocate further exploration into more informative feature representations, precisely the gap our study looks to address.

Unlike these earlier works, which either focused exclusively on deep learning or on specific feature reduction strategies, our study provides a comparative evaluation of both ensemble and shallow learning models across feature sets of varying dimensionalities. Furthermore, we contextualize our findings within deployment constraints common to IoMT systems, including computational resource limits and multi-level classification complexity.

3 CICIoMT2024 Dataset

The CICIoMT2024 dataset, released by the Canadian Institute for Cybersecurity (CIC) at the University of New Brunswick, is designed to support research on intrusion detection in IoMT environments. It was generated using a hybrid testbed composed of real and virtual IoMT devices, emulating common medical deployments such as Wi-Fi-connected monitors, infusion pumps, and patient-tracking sensors.

This dataset is notable for capturing a wide range of attack types aimed at compromising wireless IoMT communications. The attacks span multiple families:

- **Reconnaissance**: Ping Sweeps, Vulnerability Scans, OS Fingerprinting, and Port Scans.
- **Spoofing**: ARP Spoofing to poison address resolution.
- **Denial-of-Service (DoS and DDoS)**: Flooding techniques including TCP SYN, UDP, and ICMP packets.

Each attack type appears in both single-source (DoS) and distributed (DDoS) variants, simulating real-world adversarial behavior across temporal windows and volume ranges. The dataset also includes benign traffic from legitimate device operations, enabling a clear contrast between normal and malicious behaviors.

3.1 Network Topology Overview

To contextualize the data structure, it is important to understand the physical and logical setup of the data collection environment. The attack traffic was generated by four Raspberry Pi 4 Model B boards, each with varying RAM capacities (2 GB, 4 GB, and 8 GB). These devices were remotely controlled via a wireless access point connected to an iPad controller.

Traffic flowed through a network path composed of a Netgear switch, an internet gateway, and a Gigamon TAP that mirrored all Wi-Fi traffic. This TAP allowed for comprehensive packet capture without interfering with device communication. The target Wi-Fi-based medical devices were connected to a second wireless access point, simulating a typical hospital or clinic environment.

This configuration enabled the researchers to capture large volumes of packet data reflecting both legitimate and adversarial interactions. All captured traffic was stored as raw PCAP files, which were later used as input to two distinct feature extraction pipelines for analysis in this study.

4 METHODOLOGY

4.1 Overview

This study investigates how feature dimensionality impacts machine learning performance in detecting malicious traffic on IoMT networks. We used the CICIoMT2024 dataset, which has raw PCAP files collected from a multi-device wireless IoMT testbed. Two feature extraction pipelines were applied:

- **DPKT** — generates 39 low-level packet features.
- **CICFlowMeter** — generates 79 high-level flow-based statistical features.

After extraction, both datasets were cleaned, standardized, and labeled across three classification levels: binary, class, and sub-class. We trained five machine learning models on each dataset using stratified 5-fold cross-validation. After stratified 5-fold cross-validation, the processed CICFlowMeter dataset required about 25 GB of memory, while DPKT required roughly 12.5 GB. Performance was evaluated using standard classification metrics, allowing us to compare how feature richness affects both predictive accuracy and computational cost across classification tasks of increasing complexity.

4.2 Feature Extraction

Two different tools were used to extract features from the raw PCAP data:

- **DPKT (Python)**: This pipeline from the CICIoMT2024 dataset extracted low-level, per-packet features such as source/destination IPs and ports, TCP/UDP flags, IP header fields, packet lengths, and timestamps. It produced a baseline dataset holding roughly 10.7 million samples and ~2.5 GB of data.

- **CICFlowMeter (Java)**: This tool aggregates packets into bidirectional flows and extracts statistical metrics such as flow duration, inter-arrival time, and packet size variance (Lashkari, 2022). The resulting dataset was significantly larger, approximately 41 million rows and ~4 GB, offering a more context-rich view of communication patterns.

The key difference lies in granularity: DPKT captures information at the packet level, while CICFlowMeter summarizes flow-level behavior. This allows for a controlled comparison of low vs. high-dimensional feature representations.

4.3 Preprocessing

After feature extraction, both the DPKT and CICFlowMeter datasets were cleaned and structured into a common format suitable for supervised learning. Each record was labeled at three levels:

- **Binary**: 2 classes - benign vs. malicious
- **Class**: 6 classes - attack categories (e.g., spoofing, DoS, etc.)
- **Sub-Class**: 19 classes, specific attack variants (e.g., Port Scan, DDoS SYN, etc.)

The DPKT-derived dataset held approximately 10.7 million records, while the CICFlowMeter dataset expanded to over 37 million records. Interestingly, the number of benign samples dropped from 230,339 (DPKT) to 41,952 (CICFlowMeter), while attack samples increased, particularly for flood-based and connection-heavy attacks. This reflects the difference in granularity between the two tools: DPKT parses packets individually, while CICFlowMeter aggregates traffic into bidirectional flows, often fragmenting high-volume attacks into multiple flow records. Figure 1 below shows the distribution of records across attack types in both datasets. You can see that CICFlowMeter's flow-based approach increases in attack samples, especially for volumetric attacks like DoS and DDoS, compared to DPKT's packet-level parsing. This shows how the choice of a traffic analysis tool can shape the dataset's features and volume of samples.

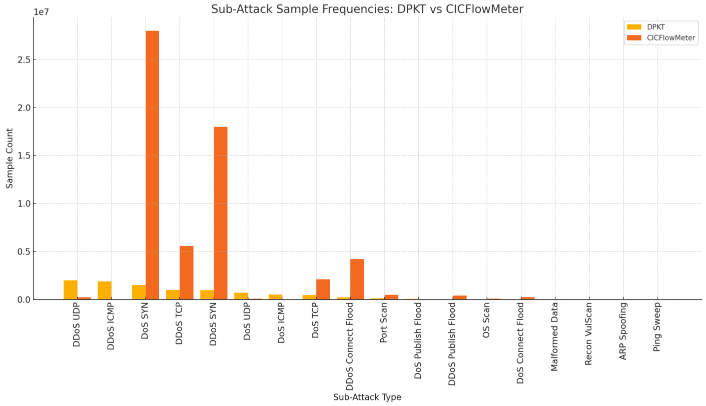

Fig. 1. Number of samples in the sub-attack task separated by dataset and sub-attack type

To address class imbalance—especially in rare sub-class labels—we applied stratified 5-fold cross-validation, ensuring that class proportions were preserved across training and validation folds. The labels were numerically encoded class IDs for model training.

5 Model Selection

We selected five machine learning models that span a range of complexity, interpretability, and computational cost: XGBoost, Random Forest, AdaBoost, MLP, and Logistic Regression.

These models were chosen to stand for a diverse set of learning strategies. XGBoost and Random Forest are ensemble tree-based models known for high accuracy and robustness to noise. AdaBoost, another ensemble method, was included for its sensitivity to weak learners and its known limitations on high-dimensional, noisy datasets. MLP, a feedforward neural network with multiple hidden layers, provides a deep learning baseline capable of capturing nonlinear relationships. Logistic Regression serves as a lightweight linear baseline, particularly useful in constrained environments.

All models were implemented using the Scikit-learn library (Scikit-learn, 2024), except for XGBoost which was implemented using the XGBoost Python package. Hyperparameters for Random Forest, AdaBoost, MLP, and Logistic Regression were directly adopted from the CICIoMT2024 paper (Dadkhah et al., 2024). These parameter sets were used to ensure consistency with prior results for comparison. For XGBoost, no configuration was provided in the CICIoMT2024 paper (Dadkhah et al., 2024), so we set the max depth to be 3 instead of 6 and included a fixed random state (Distributed (Deep) Machine Learning Community, 2025). We reduced the max depth from 6 to 3 because a lower max depth generally helps prevent overfitting and encourages the model to generalize better.

1) **XGBoost**
   - Advantages: Parallelization reduces computation time; regularization prevents overfitting (Natekin & Knoll, 2013**).**
   - Disadvantages: Memory-intensive and time-consuming despite parallelization (Rehman, 2025).
2) **AdaBoost**
   - Advantages: Focuses on misclassified data; minimizes error with sample reweighting (Vezhnevets & Vezhnevets, 2005).
   - Disadvantages: Prone to overfitting with noisy data; slower than XGBoost (Cao et al., 2012; Chen & Guestrin, 2016).
3) **Random Forest**

   Advantages: Bagging reduces data sensitivity; random subspace promotes diverse predictions (Rigatti, 2017).

   Disadvantages: Computationally intensive with large or high-dimensional data (Rigatti, 2017).
4) **Multilayer Perceptron (MLP)**
   - Advantages: Learns complex data relationships using layers and backpropagation (Marius-Constantin et al., 2009).
   - Disadvantages: Risk of underfitting or overfitting with poor parameter selection (Marius-Constantin et al., 2009).
5) **Logistic Regression**
   - Advantages: Predicts categorical outcomes efficiently using sigmoid function (DeMaris, 1995).
   - Disadvantages: Limited with complex data; struggles with many interactions (Tu, 1996).

Each model was trained using the same 5-fold stratified cross-validation strategy and evaluated across three levels of classification: binary, class, and sub-class. The goal of model selection was not to optimize peak accuracy, but rather to provide a controlled environment to assess how model architectures respond to changes in feature dimensionality. We use default parameters like the CICIoMT2024 paper (Dadkhah et al., 2024) for better comparison of the results, as look in Table 1 below.

TABLE I
THE FIVE MODELS AND THEIR DESCRIPTORS

Model	Type	Nonlinear	Parameter Source	Strengths	Limitations
Logistic Regression	Linear	No	CICIoMT 2024 Paper	Lightweight, interpretable	Weak on non-linear boundaries
Random Forest	Ensemble (Bagging)	Yes	CICIoMT 2024 Paper	High accuracy, robust to noise	Longer training time on large data
AdaBoost	Ensemble (Boosting)	Yes	CICIoMT 2024 Paper	Adaptive to errors, simple base learners	Poor scalability in high dimensions
MLP (DNN)	Neural Network	Yes	CICIoMT 2024 Paper	Captures complex patterns, flexible	Long training time, sensitive to tuning
XGBoost	Ensemble (Boosting)	Yes	Prior IDS Literature	High performance, scalable	Sensitive to parameter tuning, less interpretable

Table I: The five models used in this study and their main descriptors.

5.1 Evaluation Metrics

To assess model performance, we use a set of standard classification metrics: accuracy, precision, recall, F1 score, and log loss. These metrics are computed for each fold of a 5-fold stratified cross-validation, and results are averaged to ensure performance estimates.

- **Accuracy**: Proportion of correct predictions.
- **Precision**: Fraction of predicted positives that are positive.
- **Recall**: Fraction of actual positives that are correctly predicted.
- **F1 Score**: Harmonic means precision and recall, emphasizing balance.
- **Log Loss**: A probability-based loss function penalizing misclassifications with high confidence.

All metrics are computed per fold and on average across 5 folds. Due to class imbalance, particularly at the sub-class level, F1 score, and log loss were emphasized as primary indicators of model effectiveness and calibration.

5.2 Experiment Setup

All experiments were conducted on an HP ProLiant server running Ubuntu 22.04, equipped with an Intel Xeon processor, 755 GB of RAM, and an NVIDIA Titan X GPU. While all models were CPU-bound during training, GPU acceleration was available but was not explicitly leveraged for this study.

Model training and evaluation were implemented in Python, using Scikit-learn for Logistic Regression, Random Forest, AdaBoost, and MLP, and the XGBoost library for gradient boosting. Preprocessing and analysis were performed using standard libraries such as Pandas (Pandas, 2018) and NumPy (NumPy, 2024). For feature extraction, CICFlowMeter was executed using its original Java implementation, and DPKT-based extraction was written in Python.

RESULTS

The experimental results offer key insights into how feature dimensionality influences the performance of various machine learning models across multiple classification tasks.

5.3 Performance Gains Are Model-Dependent

Models such as Logistic Regression and Multilayer Perceptron (MLP) experienced the greatest improvement when switching from DPKT’s 39 packet-level features to CICFlowMeters' 79 flow-level features. Specifically, MLP’s F1 score for the subclass task increased from 0.109 to 0.498, while Logistic Regression improved from 0.100 to 0.423.

These gains suggest that high-dimensional features are particularly beneficial for simpler or linear models, which struggle to distinguish complex classes in sparse feature spaces.

5.4 Ensemble Models Achieve Strong Baseline Performance

In contrast, Random Forest and XGBoost demonstrated high performance even with the lower-dimensional DPKT features. Their subclass F1 scores were already near 0.996 - 0.998 using DPKT, with negligible improvement or even minor degradation on CICFlowMeter. These models appear robust to feature richness and may already capture sufficient discriminatory power from simpler inputs.

5.5 Dimensionality Greatly Affects Training Time

Training time increased significantly across all models when using CICFlowMeter features, but the degree of impact varied by architecture:

1) AdaBoost showed the most dramatic increase, with training time for subclass classification exceeding 25,000 minutes (~17 days), making it impractical for high-dimensional tasks.
2) MLP and Random Forest also experienced large slowdowns. MLP's training time on subclass classification grew from approximately 381 to 1,915 minutes, and Random Forest’s from 257 to 2,871 minutes.
3) XGBoost, however, kept reasonable training times across both datasets, completing even the sub-class task under 400 minutes using CICFlowMeter features and with DPKT features taking a little over 111 minutes.

5.6 Dimensionality Improves Generalization—But at a Cost

The addition of high-dimensional flow features generally improved classification performance for simpler models, especially on the sub-class task. However, this improvement came at a steep computational cost, particularly for models that scale poorly with dimensionality.

In resource-constrained environments, the trade-off between improved generalization and increased speed must be considered carefully. The findings visualized in Figures 2 and 3, and Tables 2 and 3 below show that:

1) XGBoost offers the best balance of speed and accuracy, making it ideal for practical deployment.
2) MLP can benefit substantially from richer features when computational resources permit.
3) Logistic Regression stays a practical lightweight baseline for binary classification tasks.

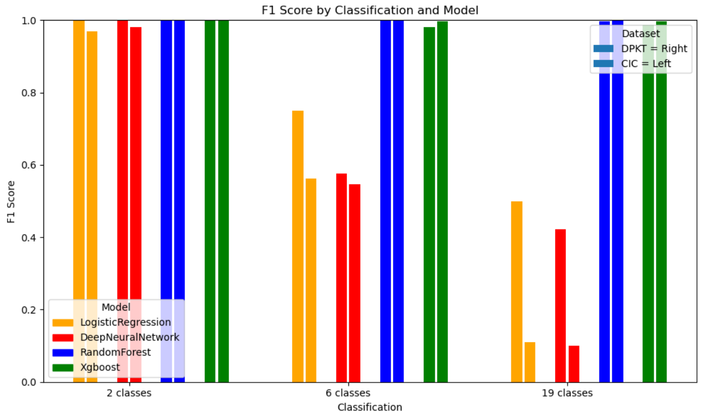

Fig. 2. Visualization of F1 scores of Models Separated by Dataset and Classification

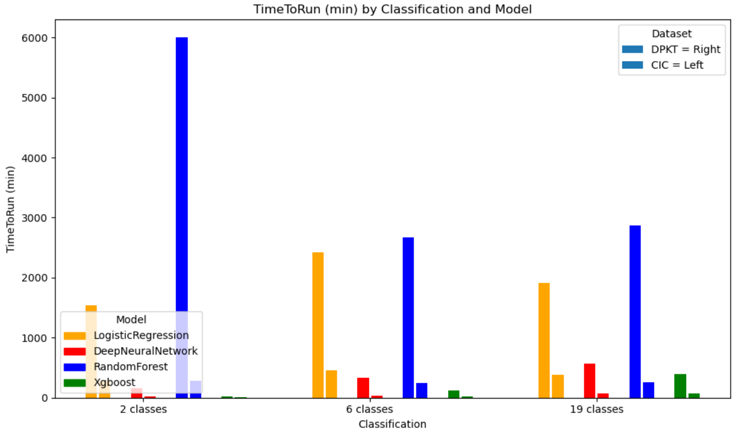

Fig. 3. Visualization of Time to Run of Models Separated by Dataset and Classification

TABLE II
F1 Score

F1 Score						
Dataset		**CIC_Flow**				
Model		**AdaBoost**	**MLP**	**Logistic Regression**	**Random Forest**	**XGBoost**
Target	**2 Classes**	0.999	0.998	0.998	0.998	0.999
	6 Classes	-	0.750	0.576	0.998	0.980
	19 Classes	-	0.498	0.422	0.996	0.987

Table II: F1 scores for various models trained on the CIC Flow dataset. A dash (–) means the model was not run for that target column/model.

TABLE III
F1 Score

F1 Score						
Dataset		**DPKT**				
Model		**AdaBoost**	**MLP**	**Logistic Regression**	**Random Forest**	**XGBoost**
Target	**2 Classes**	0.999	0.969	0.980	0.999	0.998
	6 Classes	0.998	0.562	0.546	0.999	0.997
	19 Classes	0.998	0.109	0.100	0.998	0.995

Table III: F1 scores for models trained on the DPKT dataset.

TABLE IV
F1 Score

F1 Score					
Dataset		**Original**			
Model		**AdaBoost**	**MLP**	**Logistic Regression**	**Random Forest**
Target	**2 Classes**	0.959	0.959	0.946	0.961

F1 Score					
Dataset		**Original**			
Model		**AdaBoost**	**MLP**	**Logistic Regression**	**Random Forest**
	6 Classes	0.501	0.665	0.694	0.676
	19 Classes	0.141	0.522	0.432	0.551

Table IV: F1 scores for various models trained on the CICIoMT2024 dataset.

TABLE V
Time To Run (min)

Time To Run (min)						
Dataset		**CIC_Flow**				
Model		**AdaBoost**	**MLP**	**Logistic Regression**	**Random Forest**	**XGBoost**
Target Column	**2 Classes**	25,104	1,541	152	5,997	27
	6 Classes	-	2,427	334	2,673	123
	19 Classes	-	1,915	564	2,870	390

Table V: Time to run (in minutes) for models trained on the CIC Flow dataset. A dash (–) indicates that the model was not run for that target column.

TABLE VI
Time To Run (min)

Time To Run (min)						
Dataset		DPKT				
Model		AdaBoost	MLP	Logistic Regression	Random Forest	XGBoost
Target Column	2 Classes	18	286	16	277	4
	6 Classes	12	460	36	249	22
	19 Classes	18	381	72	257	68

Table VI: Time to run (in minutes) for models trained on the DPKT dataset.

6 Discussion

The results of this study demonstrate that the impact of increased feature dimensionality on model performance is non-uniform and architecture dependent.

6.1 Benefits Are Concentrated in Simpler Models

Logistic Regression and MLP experienced the largest improvements when using CICFlowMeter's higher-dimensional feature set, particularly in complex classification tasks such as sub-class detection. These models, which rely more heavily on expressive input to compensate for limited internal complexity, showed substantial gains in F1 score. This supports the conclusion that added features can enhance discrimination power, particularly for models with limited capacity for non-linear decision boundaries.

6.2 Ensemble Models Are Saturated on Low-Dimensional Features

In contrast, Random Forest and XGBoost achieved high performance even with DPKT's 39 features, showing that these models are more efficient at learning hierarchical relationships from fewer, lower-level signals. For these algorithms, adding features offered diminishing returns, and in some cases, minor performance degradation due to overfitting or noise amplification.

This suggests that, beyond a certain point, more features do not necessarily translate into better performance, especially for already expressive models.

6.3 Training Costs Scale Non-Linearly

Increased dimensionality also introduced significant computational overhead. MLP and Random Forest, while benefiting from more features, suffered a 5 to 10 times increase in training time on the sub-class task. AdaBoost became infeasible, requiring more than 25,000 minutes to complete training on the binary task on the high-dimensional dataset.

Notably, XGBoost remained efficient in both runtime and accuracy, making it an attractive choice for deployments where low training time and high performance are both critical.

6.4 Practical Implications for IoMT Deployment

These findings highlight an important trade-off in real-world system design: while richer feature sets can improve predictive performance, they also introduce barriers to deployment in resource-constrained environments, common in IoMT and edge-based systems. Based on the findings:

1) When resources are constrained, XGBoost is the most practical choice, providing strong baseline accuracy with manageable training cost.
2) When resources are abundant, MLP can take advantage of richer features for superior generalization in complex classification tasks.
3) For simple binary tasks, Logistic Regression offers a practical, low-cost choice with respectable performance.

Ultimately, the best combination of model and feature extraction pipeline should be guided by the classification goal, available compute resources, and response time requirements of the target environment.

7 LIMITATIONS

While this study offers valuable insights into the role of feature dimensionality in ML-based intrusion detection, several limitations should be acknowledged.

7.1 Hardware and Execution Environment

All experiments were conducted on shared server hardware without containerization or process isolation. As some models were executed in parallel, resource contention at the CPU or memory level may have affected recorded training times. Consequently, comparisons involving runtime should be interpreted with caution, especially for computationally intensive models like Random Forest and AdaBoost.

7.2 Fixed Hyperparameters

Hyperparameter configurations were largely adopted from the CICIoMT2024 benchmark paper or prior literature. No exhaustive tuning or grid search was conducted to optimize model-specific parameters. This design choice ensured consistency across dimensionality comparisons but may have prevented models from achieving their best possible performance.

7.3 Synthetic Dataset Constraints

Although CICIoMT2024 is among the most comprehensive IoMT security datasets currently available, it is still generated in a controlled lab environment. The diversity and frequency of attacks, benign device behaviors, and network conditions may not fully reflect real-world IoMT deployments. As such, the generalizability of our findings regarding production systems is not guaranteed.

7.4 Feature Extraction Asymmetries

The DPKT and CICFlowMeter pipelines differ not only in dimensionality but also in data granularity and aggregation logic. DPKT generates per-packet records, while CICFlowMeter creates flow-level features by aggregating bidirectional streams. This disparity leads to differences in row counts, sample distributions, and class imbalance characteristics. While preprocessing pipelines were standardized, this structural difference may have introduced systematic biases that affect direct comparisons.

7.5 Scalability and Deployment

Models such as AdaBoost and MLP became impractical to train or deploy in high-dimensional, multi-class scenarios due to excessive memory and time requirements. This raises challenges for real-world deployment of resource-constrained IoMT devices or real-time intrusion prevention systems. Further engineering, such as model pruning, quantization, or hardware acceleration, would be needed for production use.

8 Conclusion

This study examined the effect of feature dimensionality on the performance of machine learning models in detecting malicious traffic within IoMT Wi-Fi environments. Using the CICIoMT2024 dataset, we evaluated five models—Logistic Regression, MLP, Random Forest, AdaBoost, and XGBoost—across two feature extraction pipelines of varying complexity: DPKT (39 features) and CICFlowMeter (79 features). Classification performance was assessed at the binary, class, and subclass levels.

Our results show that Logistic Regression and MLP benefit significantly from high-dimensional features, with sub-class F1 scores improving by 0.32 and 0.39, respectively. In contrast, ensemble models such as Random Forest and XGBoost achieved high accuracy even with the lower-dimensional DPKT features, gaining little or no performance improvement from added complexity.

However, feature richness came at a substantial computational cost. For example, AdaBoost needed over 17 days of training time on CICFlowMeter data for binary classification, making it impractical for real-time applications. MLP and Random Forest also experienced 5 to 10 times increases in training time. XGBoost appeared as the most balanced solution, supporting competitive accuracy while keeping training times low, even on high-dimensional data.

These findings underscore a critical trade-off in IoMT security design: while high-dimensional features can enhance accuracy, especially for simpler models, they often come with steep costs in training time and resource consumption. For constrained environments, XGBoost offers a strong default choice, providing a robust balance of speed and accuracy. Where resources are allowed, MLP offers greater upside, particularly for granular classification tasks.

Ultimately, model and feature selection in IDS design should be guided by the classification complexity, available computing resources, and deployment constraints of the target environment. This study provides actionable benchmarks to inform those decisions and encourages further work in real-world validation, feature selection optimization, and efficient model deployment for IoMT systems.

References

[1] Syamsul Arifin, A., Martadinata, T. (2025). Securing the Internet of Medical Things: A Machine Learning Approach for Cyber Threat Detection. *IAENG International Journal of Computer Science*, *52*, 901–919.

[2] Cao, J., Kwong, S., & Wang, R. (2012). A noise-detection based AdaBoost algorithm for mislabeled data. *Pattern Recognition*, *45*(12), 4451–4465. https://doi.org/10.1016/j.patcog.2012.05.002

[3] Chen, T., & Guestrin, C. (2016). XGBoost: A Scalable Tree Boosting System. *Proceedings of the 22nd ACM SIGKDD International Conference on Knowledge Discovery and Data Mining - KDD '16*, *1*(1), 785–794. https://doi.org/10.1145/2939672.2939785

[4] Dadkhah, S., Neto, E. C. P., Ferreira, R., Molokwu, R. C., Sadeghi, S., & Ghorbani, A. A. (2024). CICIoMT2024: A benchmark dataset for multi-protocol security assessment in IoMT. *Internet of Things*, *28*. https://doi.org/10.1016/j.iot.2024.101351

[5] DeMaris, A. (1995). A Tutorial in Logistic Regression. *Journal of Marriage and the Family*, *57*(4), 956–968. https://doi.org/10.2307/353415

[6] Distributed (Deep) Machine Learning Community. (2025). *XGBoost Parameters—Xgboost 1.5.2 documentation*. Xgboost.Readthedocs.Io; Distributed (Deep) Machine Learning Community. https://xgboost.readthedocs.io/en/stable/parameter.html

[7] Khaled, A. E. (2022). Internet of Medical Things (IoMT): Overview, Taxonomies, and Classifications. *Journal of Computer and Communications*, *10*(08), 64–89. https://doi.org/10.4236/jcc.2022.108005

[8] Lashkari, A. H. (2022). *Ahlashkari/CICFlowMeter*. GitHub. https://github.com/ahlashkari/CICFlowMeter

[9] Lottes, G., Kaur, M., & Ludwig, S. (2024). *Anomaly Detection in the Internet of Medical Things*. Ndsu.Edu. https://www.ndsu.edu/fileadmin/cs/REU_Posters/Anomaly_Detection_in_the_Internet_of_Medical_Things__1_-_Grayson.pdf

[10] Popescu, M.C., Balas, V.E., Popescu, L.P., & Mastorakis, N. (2009). *Multilayer perceptron and neural networks*. ResearchGate; WSEAS Transactions on Circuits and Systems. https://www.researchgate.net/publication/228340819_Multilayer_perceptron_and_neural_networks

[11] Mohammadi, A., Ghahramani, H., Asghari, S. A., & Aminian, M. (2024). Securing Healthcare with Deep Learning: A CNN-Based Model for medical IoT Threat Detection. *arXiv (Cornell University)*, 168–173. https://doi.org/10.48550/arxiv.2410.23306

[12] Natekin, A., & Knoll, A. (2013). Gradient boosting machines, a tutorial. *Frontiers in Neurorobotics*, *7*(21). https://doi.org/10.3389/fnbot.2013.00021

[13] NumPy. (2024). *NumPy*. Numpy.Org. https://numpy.org/

[14] O'Brien, G., Edwards, S., Littlefield, K., McNab, N., Wang, S., & Zheng, K. (2018). *Securing Wireless Infusion Pumps In Healthcare Delivery Organizations*. https://doi.org/10.6028/nist.sp.1800-8

[15] Pandas. (2018). *Python Data Analysis Library*. Pydata.Org. https://pandas.pydata.org/

[16] Rehman, M., Kalakoti, R., & Bahşi, H. (2025). Comprehensive Feature Selection for Machine Learning-Based Intrusion Detection in Healthcare IoMT Networks. *Proceedings of the 11th International Conference on Information Systems Security and Privacy*, 248–259. https://doi.org/10.5220/0013313600003899

[17] Rigatti, S. J. (2017). Random Forest. *Journal of Insurance Medicine*, *47*(1), 31–39. https://doi.org/10.17849/insm-47-01-31-39.1

[18] Salman, O., Elhajj, I. H., Chehab, A., & Kayssi, A. (2019). A machine learning based framework for IoT device identification and abnormal traffic detection. *Transactions on Emerging Telecommunications Technologies*, *33*(3). https://doi.org/10.1002/ett.3743

[19] Sarhan, M., Layeghy, S., Moustafa, N., Gallagher, M., & Portmann, M. (2024). Feature extraction for machine learning-based intrusion detection in IoT networks. *Digital Communications and Networks*, *10*, 205–216. https://doi.org/10.1016/j.dcan.2022.08.012

[20] Scikit-learn. (2024). *scikit-learn: Machine Learning in Python*. Scikit-Learn.Org. https://scikit-learn.org/stable/

[21] Jabbar, N., Naderan, M., & Taha, M. (2025). HybridIoMT: A Dual-Phase Machine Learning Framework for Robust Cybersecurity in Internet of Medical Things. *International Journal of Intelligent Engineering and Systems*, *18*(4), 307–321. https://doi.org/10.22266/ijies2025.0531.20

[22] Tu, J. V. (1996). Advantages and disadvantages of using artificial neural networks versus logistic regression for predicting medical outcomes. *Journal of Clinical Epidemiology*, *49*(11), 1225–1231. https://doi.org/10.1016/s0895-4356(96)00002-9

[23] Vezhnevets, A., & Vezhnevets, V. (2005). 'Modest AdaBoost'—Teaching AdaBoost to Generalize Better. *Graphicon*, *12*(5). https://www.researchgate.net/publication/239542136_

[24] Zachos, G., Mantas, G., Porfyrakis, K., Bastos, J.M.C.S , & Rodriguez, J. (2025). Anomaly-Based Intrusion Detection for IoMT Networks: Design, Implementation, Dataset Generation and ML Algorithms Evaluation. *IEEE Access*, *PP*, 1–1. https://doi.org/10.1109/access.2025.3547572

ACKNOWLEDGMENTS

The authors would like to thank Nicklaus King for his work on the computing infrastructure.

Funding statement

The project is supported by the State of Colorado through funds appropriated for Cybersecurity Law, dubbed “Cyber Coding Cryptology for State Records.” Any opinions, findings, conclusions, or recommendations expressed in this paper are those of the authors and do not necessarily reflect the views of the funding sources.

Data availability

The original data used in this study comes from the CICIoMT2024 dataset, which was released by the Canadian Institute for Cybersecurity (CIC) at the University of New Brunswick. Building on this, we created the CICFlowMeter dataset as part of our research. The code and processed CICFlowMeter dataset will be made publicly available on both GitHub and Kaggle. Links to these resources will be included upon publication.

https://github.com/Colorado-Mesa-University-Cybersecurity/The-Effect-of-Increased-Dimensionality-on-Detecting-Malicious-IoMT-Network-Traffic

https://www.kaggle.com/datasets/allenmonkey/ciciomt2024-dpkt-39-cicflowmeter-79