An Efficient Explainable Unsupervised Machine Learning Approach for Network Intrusion Detection in IoMT

Van Le[1] [0009−0001−9299−107X], Hai Minh Tran[1] [0009−0009−0186−0241], Quang Minh Tran[1] [0009−0004−2616−2152], and Tung Bui[1,2] [0000−0003−2427−5634]

[1] Bach Khoa Cybersecurity Center, Hanoi University of Science and Technology, Vietnam
{van.ltt215664, minh.thn225144, minh.tdq215616}@sis.hust.edu.vn, tungbt@soict.hust.edu.vn

[2] Corresponding author: tungbt@soict.hust.edu.vn

Abstract. The Internet of Things is progressively becoming prevalent in different industries, including medicine and healthcare. Implementing the Internet of Medical Things (IoMT) offers substantial advantages in diagnosis and treatment. Nonetheless, the IoMT-based healthcare system encounters security concerns that adversely impact the quality of therapy and directly jeopardize patient health. Many studies have employed Machine Learning to detect network intrusion on IoMT systems; however, most utilize supervised learning techniques. This research presents a detection method employing unsupervised machine learning algorithms to identify potential future attack techniques. The proposed approach incorporates the concept of Explainable AI to identify significant elements that enhance prediction accuracy. We evaluated three distinct algorithms: Kmeans, One Class SVM, and Autoencoder. The One-Class SVM model demonstrated superior performance, with an accuracy of 99.87%, a false positive rate of below 2.6%, a true positive rate of 99.98% on the CIC-IoMT2024 dataset.

Keywords: Internet of Medical Things · Intrusion Detection System · Unsupervised Machine Learning · Explainable AI · Feature Selection.

1 Introduction

The Internet of Things (IoT) has transformed numerous industries, giving rise to the Internet of Medical Things (IoMT), which integrates smart devices and systems into healthcare. IoMT's applications are broad, from remote patient monitoring to advanced diagnostics, greatly improving medical care efficiency and effectiveness. However, despite its advantages, IoMT faces significant security risks, including patient safety concerns, data breaches, ransomware, malware attacks, device hijacking, and regulatory compliance challenges. According to the 2024 Cyber Security report by Check Point [17], the global average of weekly cyberattacks in the healthcare sector has reached 1,500 cases, a 3% increase compared to the previous report. Notably, KillNet, a pro-Russian hacktivist group, has launched a large-scale operation against the US healthcare sector, conducting multiple Distributed Denial-of-Service (DDoS) attacks. These alarming statistics underscore the critical need to address the security challenges of IoMT systems.

The rising concern over IoMT security is driven by the increasing variety and sophistication of cyber threats targeting these systems. IoMT networks are vulnerable to attacks that can disrupt critical healthcare services and jeopardize patient safety. Attackers often begin with reconnaissance techniques like network, port, and OS scanning before launching targeted attacks, such as denial-of-service (DoS) or man-in-the-middle attacks that intercept and alter sensitive data. While these tactics are not new, their impact on IoMT is significant, prompting researchers to simulate attacks on IoMT systems to develop stronger security measures.

Building on the insights gained from understanding these threats, researchers have increasingly turned to machine learning as a promising approach to detect and mitigate cyberattacks within IoMT systems. The advantage of machine learning lies in its ability to adapt to and identify new and evolving threats, making it particularly well-suited for the dynamic and complex environment of IoMT. Most studies have employed supervised learning techniques to train detection models on different datasets, with some utilizing basic algorithms while others explore more advanced methods. Specifically, the effectiveness of Naive Bayes, K-Nearest Neighbor, Random Forest, AdaBoost, Logistic Regression, Deep Neural Network (DNN), Adaptive Boosting, and Support Vector Machines was examined in [1], [5], [8], [10], and [12]. Kumar et al. [4] proposed a detection framework using XGBoost, which was deployed as IaaS on the cloud side and SaaS on the fog side. In [9], directed acyclic graph-based long-short term memory (DAG-LSTM) and projected layer-based LSTM were developed. The feature section was performed in [2], [5], [6], [9], [11], and [12] to enhance the predictive capabilities of machine learning methods.

Despite these advancements, it was discovered that researchers exclusively concentrated on supervised machine learning techniques, neglecting to explore unsupervised learning approaches, which are employed to detect patterns and anomalies without prior knowledge of specific attack types. This implies it may identify novel, previously unobserved attacks that a supervised model, trained on established attack patterns, could overlook. Besides, these studies frequently employed datasets that are not specific for IoMT, including NSL-KDD, CSE-CIC-IDS2018, and ToN-IoT, or utilized datasets with limited sample sizes, such as WUSTL EHMS 2020 and ECU-IoHT. Dadkhah et al. [14] presented a comprehensive dataset collected from an IoMT testbed of 40 IoMT devices. Nevertheless, the authors conducted improper data preparation by depending on the data labels to cluster packets before training the detection model.

In this paper, we propose a model that utilizes unsupervised machine learning techniques to identify threats in the IoMT system. The objective is to develop a network intrusion detection system in the IoMT system that exhibits both exceptional precision and the ability to recognize emerging attack patterns without requiring retraining on tagged instances of these unexpected threats. The main contributions of this paper can be enumerated as follows.

- We develop an attack detection model in the IoMT systems using three unsupervised machine learning methods, namely KMeans, One-Class SVM, and Autoencoder, that can adapt to evolving threats.
- We present an appropriate way of processing the CICIoMT2024 sample dataset. Our intuitive motivation is to reduce the number of false alarms without affecting the accuracy of the attack detection model.
- We combine Mutual Information (MI) and Recursive Feature Elimination (RFE) to explain the importance of different features, which are subsequently selected to train the machine learning models, thus enhancing the overall system performance.

The remainder of this paper is organized as follows: Section 2 summarizes some related research. Our proposed method is presented in Section 3. Section 4 describes how we conducted experiments and the results demonstrating the effectiveness of the proposed method. Finally, Section 6 summarizes our research and future works.

2 Related Work

Chaganti et al. [3] proposed a PSO-DNN method to improve IoMT intrusion detection. The authors note that their approach preserves realistic normal-to-attack traffic ratios, unlike prior research that used data augmentation to balance classes. PSO is used to choose the most important aspects from network traffic and patient biometric data. After selecting features, a Deep Neural Network (DNN) model is trained to binary classify normal and attack traffic. PSO-DNN identified IoMT assaults with 96% accuracy, beating other machine learning and deep learning models in the study.

Alani et al. [13] developed XMeDNN, an explainable DNN solution for IoMT intrusion detection. Before processing the data, they removed unnecessary features, invalid samples, and random oversampling for class imbalance. The authors focused on the top five most significant features and used values to explain the model's decision-making process after 10-fold cross-validation to assure generalizability. The XMeDNN system outperformed prior research with an average accuracy of 97.578% and an F1 score of 0.97634 in 10-fold cross-validation.

Ravi et al. [7] presented deep learning-based IoMT intrusion detection that uses network flow features and patient biometric data from the WUSTL EHMS 2020 dataset. To match dataset features, the model architecture uses 1D CNNs and LSTM networks with 37 filters/memory blocks. The CNN and LSTM layers are followed by a global attention layer to maximize feature extraction. The authors use cost-sensitive learning to weigh the minority assault class more to balance the data.

In the research [6], Kilincer et al. approach for identifying IoMT network attacks and anomalies involved data preprocessing, feature extraction/selection, and classification using an MLP classifier. In preprocessing, categorical data were numerically encoded, and missing values were imputed. The most important features were selected using RFE with linear regression and XGBoost regressor kernels. The optimized MLP classifiers were tested on four IoMT security datasets: ECU-IoHT, ICU, TON_IoT, and WUSTL-EHMS.

Kumar, et al. [9] proposed two LSTM-based models proposed for cyberattack detection in IoHT networks. PL-LSTM uses a projected layer to reduce learning parameters, with output and input projectors set to 75% of hidden units and 25% of features respectively, enhancing computational efficiency. DAG-LSTM comprises two parallel LSTM models, introducing more non-linearity through its architecture of feature input, parallel LSTM layers, batch normalization, ReLU, and additional layers for processing and classification. While PL-LSTM focuses on efficiency, DAG-LSTM offers a graphical programming environment for easier model modification and debugging. Both models aim to improve cyberattack detection, with DAG-LSTM showing particular promise in handling multiple attack types in IoHT networks.

3 Proposed Methodology

In this paper, we propose unsupervised learning-based approachs for detecting network attacks in IoMT systems. The methodology, as outlined in Figure 1, illustrates the framework of our approach, which will be elaborated in the following sections.

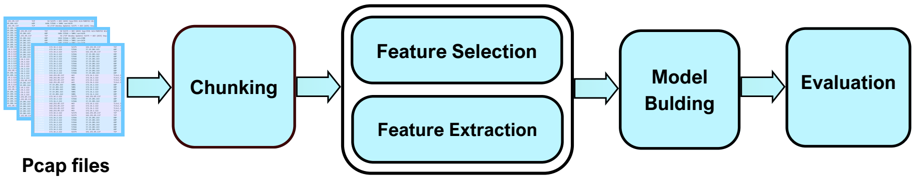


Fig. 1. Overview of the proposed method

3.1 Data Preparation

Chunking In the research [14], the authors extracted features for each packet and grouped them using a window size of either 10 or 100 packets, depending on the traffic type. However, their approach has a limitation since, in practice, traffic classification has not yet occurred, making it difficult to choose the correct window size. To address this, we first estimated the packet forwarding rate from the .pcap files in the dataset, which was approximately 13,000 packets per second. Based on this estimate, we chose a window size of 5,000 packets and used the tcpdump tool to split the .pcap files into segments accordingly.

Features Extraction To further enhance the process, we utilized DPKT to extract features from the data-link layer and CICFlowMeter to extract features from the TCP/IP layer, then combined the extracted features from both layers into a set of 97 features.

From the original 97 features, we calculated a set of five aggregation functions: mean, standard deviation, skewness, kurtosis, and median for each .csv files. Since we only needed to work with numeric features, we excluded the following 18 non-numeric features. We had a final dataset of 359 features after removing features that were identical across all samples.

We utilize the Min-max Scaling approach to normalize data in accordance with Eq. (1).

$$y = \frac{x - \min}{\max - \min} \tag{1}$$

where the *min* and *max* are the minimum and maximum values of the data to be normalized.

Features Selection Effective feature selection is crucial for improving model performance and computational efficiency. This study integrates Mutual Information (MI) and Recursive Feature Elimination (RFE) to enhance our detection models.

MI measures the relevance of each feature with respect to the target variable, capturing individual feature importance. However, MI does not account for feature interactions. To address this, we use RFE, which refines the feature set by evaluating their collective impact on model performance. MI helps in reducing dimensionality before applying RFE, making it more efficient, especially in high-dimensional datasets.

We first compute MI to identify relevant features. The mutual information between variables X and Y is given by Equation (2):

$$I(X;Y) = \int_X \int_Y p(x,y) \log\left(\frac{p(x,y)}{p(x)p(y)}\right) dx\, dy, \tag{2}$$

where $p(x,y)$ is the joint probability density function of X and Y, and $p(x)$ and $p(y)$ are their marginal densities [15].

Next, RFE iteratively removes less significant features, optimizing the subset to enhance model performance. This combined approach leverages the strengths of both methods, ensuring a well-balanced and efficient feature set.

3.2 Model training

We utilize three widely unsupervised learning algorithms, including KMeans, One-class SVM, and Autoencoder, to detect network intrusions.

Kmeans KMeans clustering is an unsupervised learning algorithm for partitioning a dataset into distinct groups or clusters. The primary objective of KMeans is to minimize the within-cluster variance, thereby ensuring that the points within a cluster are as similar to each other as possible, while points from different clusters are as distinct as possible.

One-class SVM One-class SVM, a variant of the regular Support Vector Machine, primarily serves anomaly detection and classification tasks. The specialization of this machine learning algorithm enables it to identify patterns that do not conform to the norm, which makes this technology suitable for applications such as fraud detection, network security, and quality control. The One-Class SVM differs from typical SVM in that it is exclusively trained on data that exhibits normal behavior rather than being used for classification tasks involving several cases [16].

Autoencoder An Autoencoder is a type of artificial neural network used for unsupervised learning, primarily for dimensionality reduction, feature extraction, or anomaly detection. It consists of two main components: the Encoder and the Decoder. The Encoder compresses the input data into a lower-dimensional representation, known as the latent space or bottleneck, while the Decoder reconstructs the original data from this compressed representation.

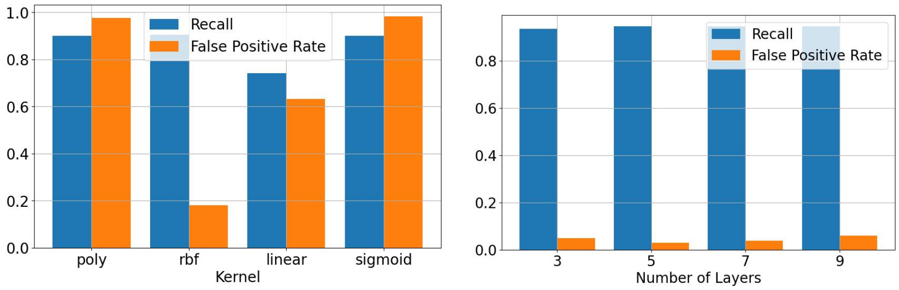
4 Experiments

4.1 Model preparation

In this section, we aim to select the most suitable kernel and the optimal number of layers for the Autoencoder to achieve the highest performance before utilizing them for feature selection.

We assessed the performance of the One-Class SVM model using four different kernel functions: polynomial, radial basis function (RBF), linear, and sigmoid. As illustrated in Figure 2(a), the results show that the RBF kernel achieved the best performance, exhibiting a high recall and the lowest false positive rate. Additionally, to reduce the computational time of the One-Class SVM, we applied Principal Component Analysis (PCA) to decrease the dimensionality of the data. This step not only streamlined the execution of the model but also preserved its classification integrity.

To determine the optimal Autoencoder architecture, we evaluated models with 3, 5, 7, and 9 layers. The performance, regarding the recall and false positive rate, is illustrated in Figure 2(b). Despite minimal variations in recall, the 5-layer Autoencoder exhibited the lowest false positive rate, marking it as the most suitable configuration for minimizing false positives in our further analysis.



(a) (b)

Fig. 2. Performance of the One-Class SVM with different (a) kernels and (b) Autoencoder layers.

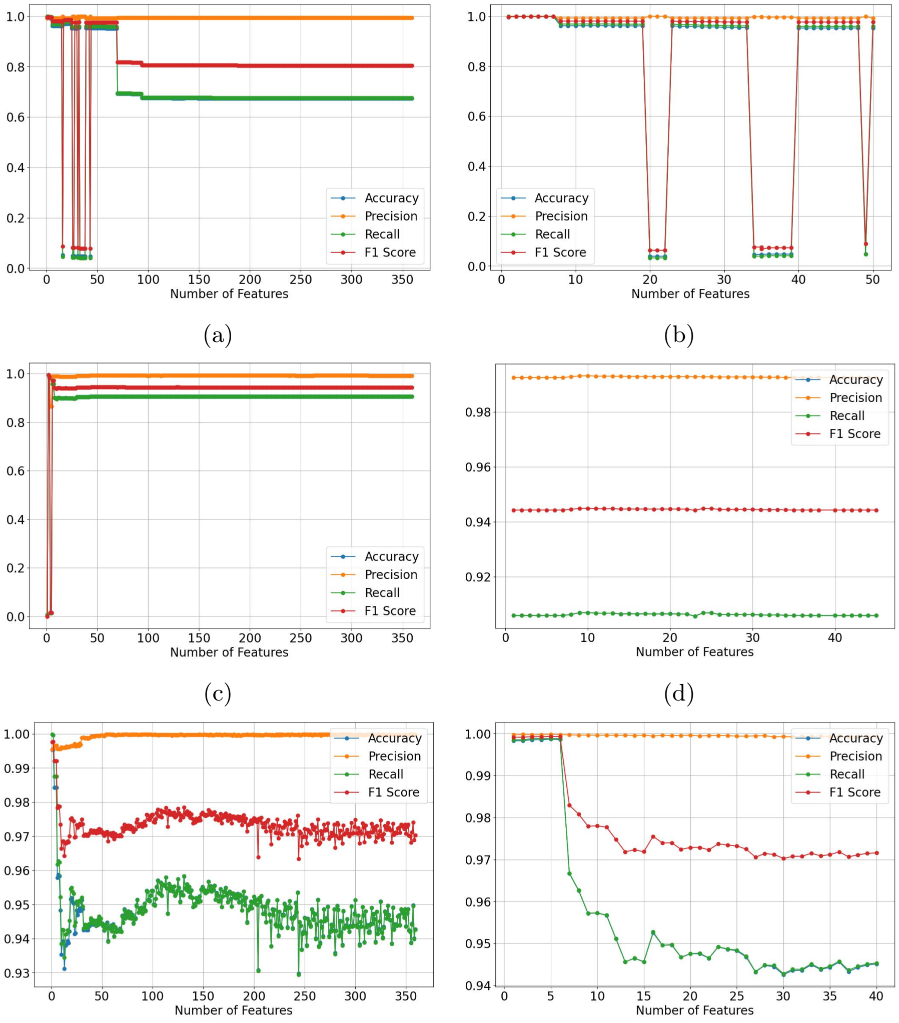
4.2 Feature Selection

Figure 3 presents the application of Mutual Information (MI) and Recursive Feature Elimination (RFE) to three models. The results from Figure 3 demonstrate the effectiveness of feature selection, showing that reducing the feature set can still deliver strong performance across the machine learning models, even outperforming the original feature set.

When applying MI, the evaluation metrics for KMeans reached their highest effectiveness when using around 50 features, then gradually declined as the number of features increased, as depicted in Figure 3(a). A similar pattern was observed with the One-Class SVM model as illustrated in Figures 3(c). For the Autoencoder, optimal performance was achieved with 45 features, as depicted in Figure 3(e). Based on these insights, we reduced the original feature set from over 300 to 50 features for KMeans, 45 features for One-Class SVM, and 40 features for the Autoencoder, ensuring that the selected features were both effective and manageable.

However, after applying MI, the feature set still exhibited a degree of dispersion, prompting us to refine the selection further by implementing RFE. With RFE, the models demonstrated more consistent and robust results, often surpassing the performance obtained using MI alone. For both KMeans and One-Class SVM, the optimal performance was achieved with just 6 and 7 features, as present in 3(b) and 3(d), while the Autoencoder performed best with a reduced set of 5 features, as illustrated in 3(f). This additional step not only streamlined the feature sets but also enhanced the overall model performance.

Building on these findings, we determined the optimal number of features for each model and compiled the list of selected features for KMeans, One-Class SVM, and Autoencoder. The results are presented in Table 1. In this table, the notation "Fwd IAT Mean (std)" indicates that the original feature "Fwd IAT Mean" was processed by calculating statistical values across the entire CSV file



(a) (b)

(c) (d)

(e) (f)

Fig. 3. Performance of different unsupervised methods when using feature selection techniques, i.e., (a) MI with Kmeans, (b) RFE with Kmeans, (c) MI with One-Class SVM, (d) RFE with One-Class SVM, (e) MI with Autoencoder, and (f) RFE with Autoencoder.

during feature extraction. Specifically, "Fwd IAT Mean (std)" represents the standard deviation of the "Fwd IAT Mean" feature computed over the file.

Table 1 illustrates features related to Fwd IAT, Flow Bytes/s, and Bwd Pkt Len Min present in all three models. In the event of a DDoS or DoS attack, these metrics change significantly, particularly with Flow Bytes/s, which may increase drastically due to an influx of packets.

Table 1. Top features as ranked by hybrid selection

Kmeans	One-Class SVM	Autoencoder
Flow Bytes/s (std)	flow Byts/s (std)	Init Fwd Win Bytes (mean)
Bwd Pkt Len Min (std)	Bwd Pkt Len Min (std)	Flow Bytes/s (std)
Fwd IAT Tot (mean)	flow Byts/s (mean)	Bwd Pkt Len Min (std)
Fwd IAT Tot (std)	Fwd IAT Tot (std)	Flow Bytes/s (mean)
Fwd IAT Mean (mean)	Fwd IAT Tot (mean)	Fwd IAT Tot (mean)
Fwd IAT Min (mean)	Fwd IAT Mean (mean)	
	Bwd Pkt Len Min (mean)	

We then present the results of the machine learning models before and after applying feature selection techniques in Table 2. This comparison highlights the impact of feature selection on model performance.

Table 2. Performance results of the models before and after feature selection.

Model	Accuracy	Precision	Recall	F1-Score
Random Forest [14]	0.9984	0.9875	0.9764	0.9819
Before feature selection				
K-means	0.4910	0.9854	0.4945	0.6585
One-Class SVM	0.8987	0.9917	0.8987	0.9429
Autoencoder	0.9243	0.9998	0.9984	0.9603
After feature selection				
K-means	0.9985	0.9987	0.9998	0.9993
One-Class SVM	0.9066	0.9933	0.9066	0.9449
Autoencoder	0.9987	0.9997	0.9990	0.9994

As shown in Table 2, after applying feature selection, the performance of the models significantly improved, highlighting the effectiveness of using MI and RFE to reduce the feature set size. Before feature selection, K-means struggled with 49.10% accuracy and an F1-score of 65.85%. However, after feature selection, it showed the most dramatic improvement, with accuracy soaring to 99.85% and the F1-score reaching 99.93%. Similarly, Autoencoder's accuracy increased from 92.43% to 99.87%, while its F1-score rose from 96.03% to 99.94%. One-Class SVM, in contrast, saw only minor gains, with accuracy rising slightly from 89.87% to 90.66%. Compared to Random Forest (RF), which remained mostly consistent, One-Class SVM lags behind in most metrics, except for a slight edge in precision. In contrast, K-means and Autoencoder surpassed RF across all metrics, achieving near-perfect results and underscoring the substantial impact of feature selection on these models.

4.3 Detection new attacks

To evaluate the ability of unsupervised algorithms in detecting novel attack types not present in the training data, we divide the dataset into three distinct scenarios:

1. **Scenario 1**: Training excludes MQTT attack samples; validation includes them. By training the model without these samples, we simulate a real-world scenario where the model may encounter MQTT attacks it has never seen before. This is particularly relevant given the growing adoption of IoT devices, which often utilize MQTT for data transmission.
2. **Scenario 2**: Training excludes DDoS SYN Flood, ICMP Flood, DoS SYN Flood, DoS ICMP Flood, Recon OS Scan, Recon Ping Sweep, and MQTT Publish Flood; validation includes these attacks. This approach allows us to evaluate the model's ability to detect attacks across different categories of DDoS, DoS, Recon, and MQTT.
3. **Scenario 3**: This scenario combines elements from both Scenario 1 and Scenario 2, excluding the same set of attack types (DDoS SYN Flood, ICMP Flood, DoS SYN Flood, DoS ICMP Flood, OS Scan, Ping Sweep, and MQTT) from the training dataset while including them in validation. This approach allows for a comprehensive evaluation of the model's performance across a broader spectrum of attack types.

In each scenario, models are trained on the designated training set and tested on the corresponding validation set to assess their effectiveness in detecting familiar and unseen attack types. The following sections provide an in-depth analysis of model performance across these scenarios, focusing on their ability to generalize to previously unobserved attacks.

Table 3. Performance Metrics Across Different Scenarios

Model	Accuracy	Precision	Recall	F1 Score
Scenario 1				
KMeans	0.9985	0.9986	0.9999	0.9992
One-Class SVM	0.8997	0.9927	0.8997	0.9439
Autoencoder	0.9986	0.9997	0.9989	0.9993
Scenario 2				
KMeans	0.9912	0.9912	1.0000	0.9956
One-Class SVM	0.8994	0.9865	0.8994	0.9410
Autoencoder	0.9985	0.9993	0.9991	0.9992
Scenario 3				
KMeans	0.9913	0.9913	1.0000	0.9956
One-Class SVM	0.8994	0.9860	0.8994	0.9407
Autoencoder	0.9983	0.9993	0.9989	0.9991

Across all three scenarios, as shown in Table 3, KMeans and Autoencoder consistently demonstrate superior and stable performance, with accuracy and F1 scores near perfect, maintaining over 99% in every case. In contrast, One-Class SVM consistently struggles, with accuracy around 89.94% and F1 scores hovering around 94%. This trend indicates that KMeans and Autoencoder are better at adapting to variations in the dataset, while One-Class SVM faces limitations when certain attack types are excluded.

Following the evaluation of the models across the three scenarios with subset 1, we then assess their effectiveness against unseen attacks using subset 2. The number of attack samples in scenarios 1, 2, and 3 within subset 2 is 4,202, 84,319, and 87,413, respectively. This diverse range of data points allows for a comprehensive analysis of the models' true positive rates across varying dataset sizes, as detailed in Table 4.

Table 4. True Positive Rate in 3 Scenarios

Model	**Scenario 1**	**Scenario 2**	**Scenario 3**
KMeans	0.9995	0.9998	0.9997
One-Class SVM	0.9976	0.9375	0.9415
Autoencoder	0.9991	0.9992	0.9991

The results in Table 4 demonstrate that all models achieve high true positive rates across the scenarios, which in turn underscores the effectiveness of unsupervised machine learning methods in adapting to the evolving threats within the IoMT system. KMeans leads with near-perfect detection, hitting 99.98% in scenario 2 and almost the same in scenarios 1 (99.95%) and 3 (99.97%). Autoencoder also performs well, maintaining 99.92% in scenario 2 and 99.91% in scenario 3. However, One-Class SVM lags, with a strong performance in scenario 1 (99.76%) but dropping significantly in scenarios 2 (93.75%) and 3 (94.15%). Overall, KMeans and Autoencoder demonstrate consistent effectiveness, while One-Class SVM struggles in more complex scenarios.

5 Conclusion

In this study, we thoroughly evaluated the effectiveness of three unsupervised learning models, including KMeans, One-Class SVM, and Autoencoder, in detecting attack types in IoMT environments. Our methodology incorporated the concept of explainable AI by using a hybrid feature selection technique, to focus on the most important features and improve model performance. The results underscore the critical role of feature selection in anomaly detection, demonstrating that irrelevant or redundant features can reduce the effectiveness of unsupervised models. Utilized algorithms exhibited strong performance in identifying previously unseen attack types, indicating their potential as viable solutions for real-world IoMT security challenges where new threats are continually emerging.

In the future, we will explore hybrid models that leverage the strengths of various algorithms to further improve detection capabilities. Expanding the scope