EAI Endorsed Transactions
on Internet of Things
Research Article EAI.EU

An Adaptive Intrusion Detection System for Securing the Internet of Medical Things Using Deep Learning

Abdullah M. Albarrak*

Computer Science Department, College of Computer and Information Sciences, Imam Mohammad Ibn Saud Islamic University, Riyadh, Saudi Arabia

Abstract

INTRODUCTION: The Internet of Medical Things (IoMT) has expanded rapidly, with a growing number of medical devices becoming interconnected and increasingly integral to healthcare delivery. However, this expansion has also introduced significant cybersecurity risks, making IoMT networks vulnerable to sophisticated cyber-attacks that threaten patient data confidentiality and system reliability.
OBJECTIVES: This study aims to develop a robust intrusion detection framework capable of accurately identifying both known and unknown cyber-attacks in IoMT environments while minimizing false positives and false negatives.
METHODS: The proposed framework employs a Capsule Neural Network (CapsNet) to effectively capture spatial hierarchies and part–whole relationships in network traffic data. Additionally, the Theory of Association (TOA) is utilized for batch-size hyperparameter tuning to enhance learning efficiency and detection performance. The model is evaluated using standard performance metrics to assess its effectiveness in detecting malicious traffic. RESULTS: Experimental results demonstrate that the proposed Intrusion Detection System (IDS) achieves an accuracy of 98.37%, precision of 98.57%, recall of 98.17%, and an F1 score of 98.37%. These results indicate strong real-time detection capability with minimal false positives and false negatives.
CONCLUSION: The findings highlight the effectiveness of integrating deep learning techniques, particularly CapsNet with TOA-based optimization, in strengthening cybersecurity for IoMT networks. The proposed IDS provides a secure and efficient solution for protecting healthcare data and ensuring patient confidentiality, offering a promising approach to enhancing the security and performance of healthcare IoMT systems.

Keywords: Intrusion Detection System, Internet of Medical Things, Capsule Networks, Teamwork Optimization Algorithm.

Received on 19 September 2025, accepted on 10 November 2026, published on 19 February 2026

doi: 10.4108/eetiot.10326

*Corresponding author. Email: amsbarrak@imamu.edu.sa

1. Introduction

The IoMT has expanded quickly, with more networked medical devices becoming key in healthcare. These devices provide crucial services, including pacemakers, insulin pumps, and wearable health monitors, to track and treat patients [1]. However, as dependence on these connected devices increases, so does the risk of cyberattacks. Since these gadgets store and transmit sensitive patient health data, they become prime targets for hackers. Protecting IoMT networks is vital not only for patient safety but also for maintaining the integrity and confidentiality of individuals' health information [2]. The rise in connected medical devices has also made IoMT networks more susceptible to cyber threats [3]. Securing these devices is essential to ensure patient safety, data accuracy, and the protection of personal health information in [4]. As cyberattacks grow more sophisticated, traditional security measures may be inadequate, making the development of advanced IDS for IoMT a pressing concern [5].

There are several reasons why IoMT networks are vulnerable to attacks, for example, the limited processing power of medical devices in the limited security protocols embedded in the devices, and always-on connectivity [6]. Most of the devices operate with or without encryption and persistent firmware updates with minimal security assessment, opening them to numerous attacks [7]. The large number of IoMT devices, commonly spread out among hospitals and patient residences in Urban and Rural areas of makes real-time monitoring and security even more challenging [8]. Increased use of IoMT devices requires increased demand for new IDS techniques in securing such infrastructure [9].

Various approaches have been suggested for IoMT network security in including signature-based IDS, anomaly detection, and ML-based methods [10]. Conventional signature-based IDS systems utilize pre-defined attack signatures and are ineffective against new or zero-day attacks [11]. Anomaly detection systems, being more adaptive, are bound to generate high false-positive rates due to the variability of medical data [12]. ML algorithms, like SVM and RF, are also widely used but do not discover the complex and hierarchical dependencies among features, limiting their ability in real-time IDS in IoMT infrastructure [13].

The proposed framework addresses the above drawbacks with the help of CapsNet, a DL that learns part-whole relationships and spatial hierarchies in data. Through the combination of dynamic routing and DL, it discovers the known and unknown attacks in IoMT networks. Not only does it decrease false positives, but it also makes it more effective to learn adaptive threats. The novelty of the new research is in its ability to handle complex attack patterns and intrude more precisely and timely manner. The key contribution of this paper:
Due to the dynamic evolution of cyber threats, an adaptive IDS using CapsNet has been proposed for efficient malicious traffic detection in IoMT networks.

The study involves the TOA for hyperparameter tuning, which optimizes key parameters like batch size for greater IDS detection accuracy and performance.

By learning spatial hierarchies within IoMT network data, the framework can efficiently identify both known and unknown attacks, which was beyond the capability of traditional IDS.

The proposed solution is set to address specific challenges posed by IoMT networks such as network diversity and device heterogeneity, thereby promoting the approach as a viable tool for improving healthcare data security.

The paper structure follows: Section 2 reviews existing IoMT IDS literature. Section 3 describes the proposed CapsNet-based methodology. Section 4 presents results, and Section 5 concludes with future work.

2. Literature Survey

Zachos et al. presented an IDS for IoMT anomaly detection has been presented to safeguard these systems. It works using host and network factors to download log files and traffic data from an IoMT network under computational constraints from the IoMT devices. Six ML models are analyzed under different conditions for anomaly detection, with the most efficacious selected to detect attacks on IoMT networks; this will, thereby, secure and enhance healthcare systems [14]. Alsoufi et al. carried out an SLR describing the use of DL methodologies in anomaly-based IDS to secure IoT environments. The study comprises 26 relevant publications from 5 databases and seeks to establish how DL techniques help in IoT security issues, more specifically, in the detection of zero-day attacks. It shows that supervised deep methods outperform unsupervised and semi-supervised methods and reveal important information relating to how data types and learning criteria affect the performance of IDS. [15].

Toldins et al. present that cyberattacks and hacking are on the rise, and the need for strong IDS has never seemed to be more urgent. This study proposes a novel intrusion detection method based on multistage deep learning and image recognition. Network features are converted into four-channel images. The system is trained and tested using the pre-trained deep learning model ResNet50 so that the precision of anomaly and network attack detection can be improved to a great extent, which finally leads to a stronger intrusion detection solution [16]. According to Nayak et al. proposed an IoMT framework was proposed to take into account security issues within health systems. The aim is to increase predictability in decision-making processes and facilitate the security of patient information so that it is not made public by some vulnerability in IoMT devices. Through the amalgamation of Bayesian optimization with equipment communication security, privacy is being considered in this research to avoid unauthorized access or physical injury to patients [17].

Chaganti et al. present a Particle Swarm Optimization (PSO) Deep Neural Network DNN for IDS in IoMT. This system takes the problem of cybersecurity in IoMT devices, detecting cyber-attacks with 96 percent accuracy in patient sensing data. It compares ML and DL techniques, supporting that deep learners have a slight edge over ML in IDS, thus dealing with healthcare applications for IoMT [18]. Ullah et al. present an integrated DL to detect cyberattacks in the Internet of Vehicles (IoV) that interconnects vehicles through IoT networks. The model combines Long Short-Term Memory (LSTM) techniques. The model was tested on combined Distributed Denial of Service (DDoS) and car-hacking datasets, detecting DDoS attacks with an accuracy and car-hack attacks with an accuracy. It supports the successful application of the proposed method in improving the security of the IoV [19].

Alalhareth and Hong present a fuzzy self-tuning LSTM-based IDS for IoMT security in healthcare operations. Their model considers the number of epochs as a variable that can be increased or decreased, along with early stopping as a means to deter overfitting. From extensive experiments, the system has proven to have high accuracy, low false alarm rates, and efficient IDS. It provides a better solution than the other available IDS models and fills the gap of static DL avenues [20]. Vishwakarma and Kesswani present the two-

phase classifications of IDS. The classification into four data types-nominal, integer, binary, and float-using Naive Bayes classifiers and majority voting. The work on the extraction of benign data uses an unsupervised method called an elliptic envelope. Their method has shown the best accuracy on multiple datasets, much better than the traditional ways of IDSs [21].

According to Si-Ahmed et al. application of ML-based IDS to threats faced by IoMT systems. It is mainly breaches of data and attacks due to its heterogeneous nature and limited resource constraints that demand security for IoMT. The paper presents a three-layer IoMT architecture, analyzes threats at all the levels, and reviews ML-based IDS solutions at those levels for their advantages and disadvantages and concerns that the author believes are worthy of being considered in future research [22]. Vijayakumar et al. presented the virtues of the newest ECU-IoHT dataset, an artificial intelligence network of cyber-attack detection for the Internet of Health Things (IoHT). It uses anomaly detection methods for security management events. The system carries an accuracy, an area under the receiver operating characteristic curve (AUC) [23].

Sun et al. introduced a PSO algorithm combined with AdaBoost for detecting malware on health app platforms. Using NSL KDD as the database, the system recognized 12 features of importance for IDS and successfully categorized the different types of attacks. AdaBoost marks the highest recall value, meaning that it performs well in IDS. The study makes a case for machine-learning-based IDS to foster security, minimize cost, and maximize patient outcomes on the IoMT [24]. Talukder et al. present an advanced IDS methodology for WSNs by merging interior ML mechanics. It synthesizes minority instances for resampling and eliminates Tomek links to enhance detection accuracy. Their work considers this approach with WSN-DS data. These depict the model in overcoming barriers relating to data imbalance and successfully increasing IDS in WSNs [25].

2.1 Problem Statement

Conventional IDS tends to be static, not effective against new and novel threats, and incapable of keeping up with the changing nature of IoMT networks [15]. These systems also have high FP rates and are not very scalable [21]. To develop an adaptive IDS that utilizes CapsNet for precise classification and hyperparameter optimization, employing the TOA to enhance the model's detection capabilities, reduce false alarms, and increase its robustness against various attack patterns in IoMT contexts. The proposal aims to enhance security, provide privacy, and keep patient data safe against advanced cyber-attacks on IoMT devices[23].

3. Proposed Methodology

The proposed methodology consists of several key steps designed to enhance the detection and reliability of cyber threat detection in heterogeneous situational IoT contexts in Figure 1. The initial step of the methodology is Data Collection using the Botnet Internet of Things (BoT-IoT) dataset. The second step is Data Preprocessing, which includes the treatment of missing values (where indicated) within the dataset, normalization of the training and testing data, and finally splitting the collected data into a training and test set. The third step optimizes hyperparameters with the TOA. The batch size was optimized to improve the performance of the CapsNet neural network with a focus on accuracy, which minimizes false positives and false negatives in the detection of IoMT.The final step is Feature Extraction and Classification with CapsNet, which has excellent handling of both benign and adversarial traffic. This integrated methodology offers high accuracy, reliability, and adaptability while dealing with a vast array of devices and types of attacks experienced during IoT contexts.

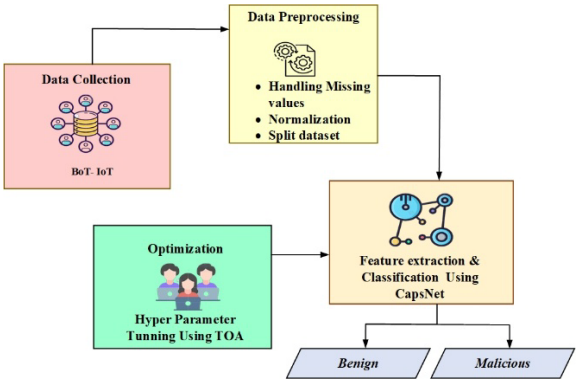

Figure 1. Overall Proposed Methodology

3.1 Data Collection

The data employed from the BoT-IoT dataset, which includes both legitimate and botnet traffic samples, as well as a variety of attack patterns used against different network protocols. The dataset is organized to make it easy to label and analyze the traffic data, making it especially useful for research work. It is open access for non-commercial educational use, provided users properly cite the source authors' papers if they make use of the data. This dataset has significant research value in IoT security to investigate the vast variety of IoT devices and the associated cyber threats.

3.2 Preprocessing

Data preprocessing involves a series of essential steps to prepare a dataset for model training. Missing values are handled either by imputation or by removing data if the missing part is significant. The next step is normalization, where features are scaled to a specific range, usually using min-max scaling or Z-score normalization, to improve performance. Finally, creating a training set, a validation set, and a test set should be done effectively to avoid overfitting

and to evaluate the model on test data. These preprocessing steps are crucial for preparing the dataset for further analysis and model development.

3.2.1 Handling Missing Values

The common challenge in record linkage is to fill in the missing values, or one might also opt to weed out the missing values. Imputation includes replacing missing data with, say, a statistical metric measure of central tendency of the respective feature. The mean imputation in Eq (1):

$$X_{\text{imputed}} = \mu(X) \tag{1}$$

where, X is a variable with missing values and $\mu(X)$mean is computed by considering only the values present for X. One can always afford to delete the missing records or even the missing columns if the ratio of missing data is too large to maintain data integrity and train a well-fitting model.

3.2.2 Normalization

Scaling features ensures all features have equal importance while learning, which is a crucial requirement for DL like CapsNets. Min-Max scaling puts feature values in the range of [0,1] based on the formula in Eq (2):

$$X_{\text{scaled}} = \frac{X - \min(X)}{\max(X) - \min(X)} \tag{2}$$

where, X indicates the original feature, $\min(X)$ and $\max(X)$ specifies the minimum, maximum of feature. Meanwhile, the Z -score, where features are stretched according to their mean and standard deviation in Eq (3):

$$X_{\text{standardized}} = \frac{X - \mu(X)}{\sigma(X)} \tag{3}$$

Where, $\mu(X)$ is mean and $\sigma(X)$ is the standard deviation of X. These methods put the features on the same scale, thus increasing their efficiency and efficacy when used in models like CapsNets.

3.2.3 Data Splitting

Repeating Data Science's very important step of splitting a dataset into three sets so that the machine can get enough data for training, validation, and testing. The Training Set is the actual training data from which patterns are learned by the model. The Validation Set is to tune the hyperparameters to ensure it does not overfit and generalizes well to unseen data. Finally Test Set is to test for hidden data, thereby genuinely assessing its prediction ability. The dataset is divided by using the Eq (4):

$$X_{\text{train}} + X_{\text{val}} + X_{\text{test}} = X_{\text{total}} \tag{4}$$

where X_{total} is the entire dataset, and the subsets are usually divided based on specific ratios

3.3 Feature Extraction and Classification using Capsule Net

The multi-layered, CapsNet would be applied to classify network traffic in Figure 2. Traffic data is received by it at the input layer after preprocessing and is convolutional layers to the data. The features are transferred to the primary capsule layer, where the capsules symbolize lower-level features. Dynamic routing lends its support by directing information from one capsule to another based on the extent to which they match; this enables higher-level capsules to combine lower-level information into more specific representations. Finally, the output layer uses softmax activation to categorize traffic into types like benign and malicious.

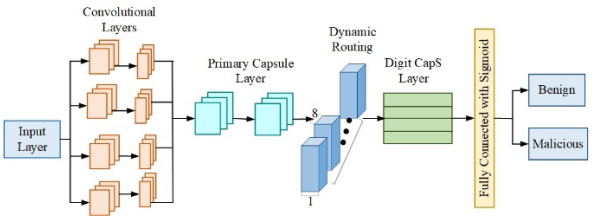

Figure 2. Capsule Network Architecture for Network Traffic Classification

3.3.1 Input Layer

The preprocessed data are given as the input layer, which is divided into sub-packets like Convolutional layer, Primary Capsule Layer, Dynamic Routing, Digit CapS Layer. It finally goes to the fully connected sigmoid for classifying the Benign and Malicious Traffic in the Capsule Network.

3.3.2 Convolutional Layers

The Convolutional Layers use filters on input data to glean low-level features such as patterns and edges in network traffic. Such layers generate feature maps that reflect the simple structures. The processed feature maps then get sent to yet another layer primary capsule in Eq (5).

$$F_{conv} = W_{conv} * X_{input} + b_{conv} \tag{5}$$

Where, W_{conv} is a convolutional filter, b_{conv} is a bias term.

3.3.3 Primary Capsule Layer

The set of neurons or capsule groups features for the function detected by the convolutional layers. The capsules encode such features as position, orientation, and probability of occurrence of a certain feature in the input data. The output

of each capsule represents a vector of instantiation parameters for the feature detected in the data by using the Eq (6).

$$C_1 = W_{\text{capsule}} * F_{\text{conv}} + b_{\text{capsule}} \tag{6}$$

Where, C_1 is the primary capsule layer, W_{capsule} is a weight matrix, and b_{capsule} is bias term.

3.3.4 Dynamic Routing

A dynamic routing allows the transfer of information between capsules depending on the spatial and hierarchical relations of the features. The dynamic routing algorithm will update coupling coefficients on the degree of agreement of lower-level capsules with the higher-level ones in Eq (7).

$$\mathrm{v}_i^{(\text{higher})} = \sum_j c_{ij} \mathrm{v}_j^{(\text{lower})} \tag{7}$$

Where, $\mathrm{v}_i^{(\text{higher})}$ is the output vector of the higher-level capsule, c_{ij} is the routing weight, $\mathrm{v}_j^{(\text{lower})}$ is the output vector from the lower-level capsule.

3.3.5 Digit Capsule Layer

It processes high-level features, combining and integrating vectors representing lower-level features from a variety of sources, to carve out complex relations and patterns on the network traffic data. This layer is capable of encapsulating the part-whole relationships and discriminating among intricate traffic patterns, such as attack types in Eq (8).

$$\mathbf{v}_i^{(\text{final})} = \sum_j c_{ij} \mathbf{v}_j^{(\text{higher})} \tag{8}$$

Where, c_{ij} is the dynamic routing coefficient, and $\mathbf{v}_j^{(\text{higher})}$ This is the output from the higher-level capsules.

3.3.6 Fully Connected Layer with Sigmoid

The layer produces probabilities for each class, e.g., benign or malicious traffic. Using either a softmax or sigmoid function, the output is guaranteed to have a value between 0 and 1, after which these values can be interpreted as class probabilities in Eq (9).

$$\hat{y}_i = \frac{1}{1+e^{-z_i}} \tag{9}$$

where $\hat{y}_i$ is the predicted class probability for class i, and z_i Is input to the sigmoid function.

3.4 Hyperparameter Tuning Using Theory of Association

CapsNet-based IDS is optimized using the TOA to fine-tune the hyperparameters. TOA is specifically used to optimize the batch size, a highly critical hyperparameter affecting the performance of the model. TOA borrows from collaborative concepts in teamwork, where agents work together to discover the best collection of hyperparameters. TOA, through collaboration, allows the system to learn better about sophisticated attack patterns within IoMT networks. The application of TOA guarantees the CapsNet model is running at maximum capacity, reducing errors while maximizing its detection power. The TOA is inspired by team members working together in pursuit of a shared objective. The interaction and relationship between team members, including supervision by the supervisor and sharing information, are used as metaphors for solving the optimization problem of identifying the optimal hyperparameter in batch size. (B)R Training the CapsNet model.

3.4.1 Population Initialization

In the TOA, several agents represent a population of possible solutions to the optimization problem. For example, each agent may propose different hyperparameter values for the CapsNet model, i.e., batch size, learning rate, or any other must-have variable. The suggested solution finds its representation in a matrix wherein rows stand for individual agents, and the columns stand for the hyperparameters under consideration. The population is represented by the matrix, with each entry of the matrix giving the value of a given hyperparameter as suggested by one of the agents. If these hyperparameter values are optimized, it means that agents learn in collaboration and wish to exchange information to come up with the set of hyperparameters that will best support the improvement of an application's performance in Eq (10):

$$Y = \begin{bmatrix} Y_1 & Y_{1,1} & \cdots & y_{1,d} & \cdots & y_{1,m} \\ \vdots & \vdots & \ddots & \vdots & & \vdots \\ y_i & Y_{i,1} & \cdots & y_{i,d} & \cdots & y_{i,m} \\ \vdots & \vdots & & \vdots & \ddots & \vdots \\ y_N & y_{N,1} & \cdots & y_{N,d} & \cdots & y_{N,m} \end{bmatrix}_{N\times m} \tag{10}$$

Where, y stands for population matrix, y_i is for i^{th} member of the team, N is the number of members, and m is the total number of problem variables.

3.4.2 Objective Function Evaluation

The proposal for hyperparameters by each agent is checked by an objective function, which evaluates the performance of the CapsNet model trained with some given hyperparameters. The objective function could be the performance for all agents is calculated and stored in a vector. It means the algorithm can evaluate at every step the hyperparameters proposed by every agent; this evaluation guides the updates in the optimization in Eq (11):

$$F = \begin{bmatrix} F_1 & F(X_1) \\ \vdots & \vdots \\ F_i & F(X_i) \\ \vdots & \vdots \\ F_N & F(X_N) \end{bmatrix}_{N\times 1} \tag{11}$$

where F is the objective function vector, and F_i is the value of the i^{th} team member.

Step 1: Supervisor guidance

During a first update step the members need to be instructed by the supervisor. The supervisor imparts information to some members and reports to some others so that the director has a general understanding and can guide some members toward the realization of a particular purpose. This kind of update is simulated by means of the following sets of equations (12) and (13) in the TOA.

$$X_i^{S1}: x_{i,d}^{S1} = x_{i,d} + r \times \left(S_d - I \times x_{i,d}\right) \tag{12}$$

$$X_i = \begin{cases} X_i^{S1}, F_i^{S1} < F_i \\ X_i, \text{else} \end{cases} \tag{13}$$

where X_i^{S1} refers to a new status for i^{th} team members coming directly from the supervisor's changes, F_i^{S1} is the objective function value, $x_{i,d}^{S1}$ means new values for d problem variables, I is the index for updating, and r is a random number.

Step 2: Information sharing

An attempt to utilize the data from one team member to another is performed better than themselves in the hope of seeing an improvement in the performance level. Some rules that represent the Updating of Members Stage, in the second stage of the proposed TOA, are in Eq (14), (15) and (16).

$$X^{M,i}: x_d^{M,i} = \frac{\sum_{j=1}^{N_i} x_{j,d}^{g,i}}{N_i} \tag{14}$$

$$X_i^{S2}: x_{i,d}^{S2} = x_{i,d} + r \times \left(x_d^{M,i} - I \times x_{i,d}\right) \times \text{sign}(F_i - F^{M,i}) \tag{15}$$

$$X_i = \begin{cases} X_i^{S2}, F_i^{S2} < F_i \\ X_i, \text{else} \end{cases}, \tag{16}$$

where $X^{M,i}$ is the mean of those team members, $F^{M,i}$ is the value of the objective function, $x_{j,d}^{8,t}$ is the value of d variable, X_i^{S2} is the original state for i^{th} team member based on the second stage, and F_i^{S2} is the objective function value.

Step 3: Individual activity

During the third stage of optimization, work is done on perturbing slightly the current state for an envisaged solution. The current value is multiplied by a factor that involves a random number and helps explore new solutions that could yield improvements in the value of the objective function. The update can be expressed by Eq (17)

$$X_i^{S3}: x_{i,d}^{S3} = x_{i,d} + (-0.01 + r \times 0.02) \times x_{i,d} \tag{17}$$

where X_i^{S3} represents a new state, and F_i^{S3} represents the objective function value.

Pseudo code for TOA

Start TOA
Step 1: Input problem information
Define the problem variables and the objective function
Define Variables: batch size (B)
Step 2: Set the number of team members (N) and iterations (T)
N = Number of agents (team members)
T = Number of iterations
Step 3: Generate the initial population matrix with random values
Initialize Population Matrix X
Step 4: Evaluate the objective function for each agent
for each agent i in range $(1, N+1)$:
Evaluate the Objective Function (X_i) using Eq (11)
For each iteration t (1 *to* T)
for t in range $(1, T+1)$:
Step 5: Update supervisor
Evaluate the Select Best Agent
S = Select Best Agent (X)
Step 6: For each agent $\boldsymbol{i}$ ($\mathbf{1}$ ***to*** $\boldsymbol{N}$)
for i in range (1, N+1):
Stage 1: Supervisor Guidance
Update X_i based on the First stage using Eq (12) and Eq (13)
Stage 2: Information Sharing
Determine better team members and N_i for i^{th} team member.
Calculate Eq (14).
Update Xi based on the second stage using Equations (15) and (16).
Stage 3: Individual Activity
Update X_i based on the third stage using Eq (17)
End TOA

4. Result And Discussion

The IDS is extremely effective for detecting malicious traffic in IoMT networks. It has high precision with an accuracy of 98.37%, precision of 98.57%, and recall of 98 .17%. The proposed work highlights the model's capacity for effectively discriminating between benign and malicious traffic, and

relying on metrics such as precision and recall, the model has shown low false positives and false negatives. The confusion matrix also reflects that the model is classifying both benign and malicious data well, and the ROC curve and precision-recall curve also demonstrate the strength of the proposed approach. The IDS works well and will provide security to IoMT networks, and can also be deployed in real-time. The confusion matrix, ROC curve, and precision-recall curve maintain the argument statistic about being robust with very low FP and FN rate and good generalization on unseen instances.

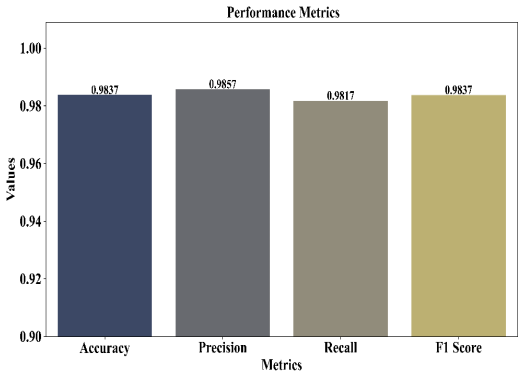

Figure 3. Performance Metrics

Figure 3 shows the IDS with parameters for measuring its being measured for performance. It has an accuracy of 0.9837; while this was a general indicator of the working of prediction, precision shows us that the detection system is extremely competent at detecting actual instances of maliciousness while not raising too many false alarms by way of false positives (FP). A recall of 0.9817 means the model identifies with great efficacy malicious instances with false negatives (FN). F1 score of 0.9837 shows a balance in terms, which contributed to the functioning in detecting malicious traffic well with very few errors.

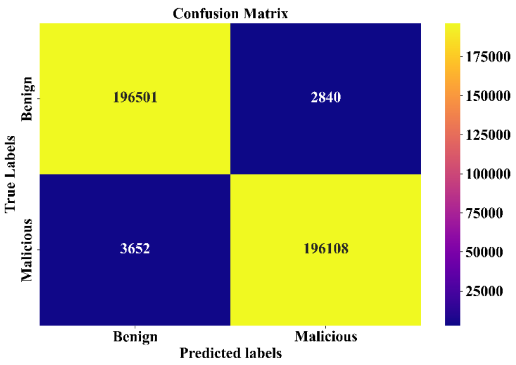

Figure 4. Confusion Matrix

Figure 4 displays true and predicted labels compared for a measure of model performance. The model was able to correctly classify 196,501 benign instances and 196,108 malicious instances as true positives (TPs). However, 2,840 benign instances were wrongly classified as belonging to the malicious class, while 3,652 malicious instances were classified incorrectly as benign. The color gradient represents the frequency of predictions, with yellow showing the highest counts. The model did very well with the identification of both the benign and malicious traffic.

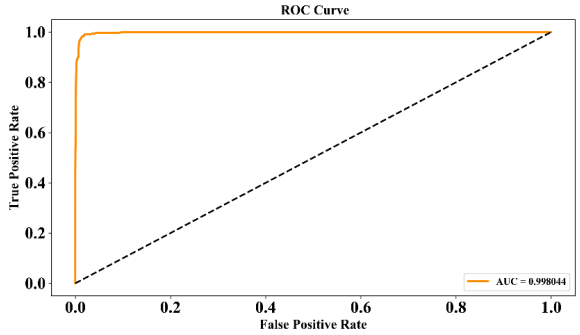

Figure 5. ROC Curve

Figure 5 expresses the ROC curve for the IDS. The ROC curve plots the TP rate against the false positive rate. The ROC curve speaks greatly about the model, with TP indicators reaching near 1 very fast, with FP indicators. This demonstrates the model's ability to detect malicious traffic with very few false alarms. The AUC value of 0.998044 indicates almost perfect performance, as values close to 1 indicate the model's ability to effectively distinguish between traffic that is benign or malicious.

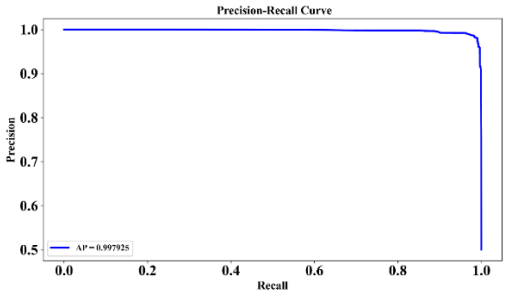

Figure 6. Precision-Recall Curve

Figure 6 represents the IDS, exhibiting a trade-off at numerous recall values. The curve remains very close to the top right corner, depicting that the idea retains high precision as well when increasing recall. The value of AP (Average Precision) being 0.997925 emphasizes the capability of this model and underlines how it correctly labels malicious traffic while at the same time hardly ever classifying benign traffic

as malicious, i.e., with very few false positives. Hence, high recall and precision will allow this model to detect benign and malicious traffic with the least amount of error.

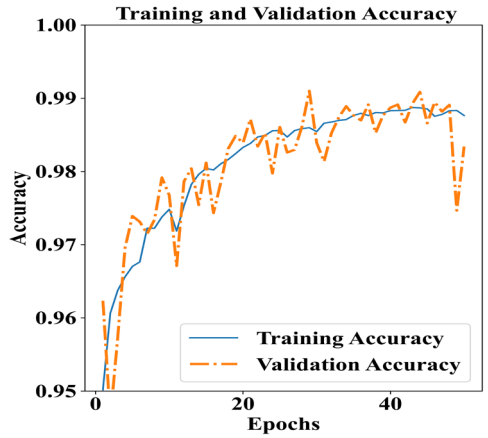

Figure 7. Training and Validation Accuracy

Figure 7 shows the blue curve; the training accuracy is rising steadily and has nearly reached 0.99 after about 45 epochs. The validation curve (comprising orange dashed lines) shows an increasing accuracy that peaks at about 0.98. Having similar trends for both accuracies confirm an effective model learning without much overfitting. The diets are apt to slight overfitting in the later epochs, as indicated by the small gap existing between these two curves, but generally, the model performs excellently, with training accuracy hitting 0.99 and validation accuracy nearing 0.98.

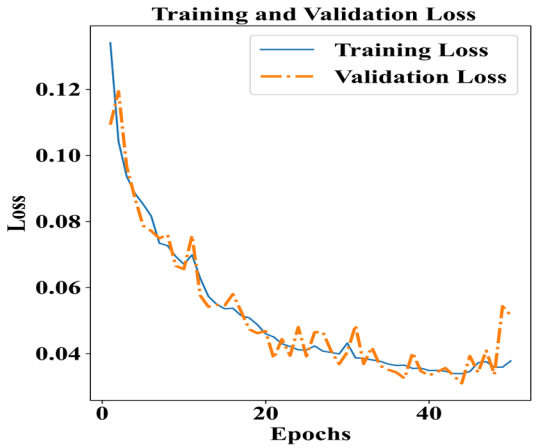

Figure 8. Training and Validation Loss

Figure 8 shows Training loss, an anomalously huge value of 0.12 followed by a steady decline, represented by the solid blue line, which almost reached 0.04 at the 45th epoch. It was almost a reverse scenario with the validation loss, where it decreased for some epochs and later began to fluctuate. Both losses also experienced a good diminution during the early stages of training, manifesting that the model generally learned very well, with performance improving steadily. There exists a very small gap between training and validation losses, which can be indicative of the model generalizing well on unseen data or not being significantly overfitted.

5. Conclusion

An adaptive IDS for safeguarding IoMT networks, using CapsNet, along with the TOA for hyperparameter tuning. The model presented in this paper demonstrated ideal capabilities for the detection of both known and unknown cyber-attacks in real-time with superior performance in relation to accuracy, precision, recall, and F1 score. The leverage of CapsNet structure that detects complex spatial hierarchies in the data, combined with TOA's optimization for batch size, was shown to catapult the success and reduce the false positives and false negatives. With the increasing complexity and unpredictability of the threat landscape in healthcare, this coordination of technologies represents a vaccination against cyber threats.

The results show that the model is very effective, considering that it had an accuracy, precision, and recall of 98.37%, 98.57%, and 98.17%, respectively. Future work could involve testing the model using actual IoMT datasets specifically designed for to increase its relevance in the healthcare setting. Upgrading the algorithm to be based on sophisticated machine learning or hybrid models could also help increase detection performance while deploying IoMT technologies in a wider context will help assess if the model can truly function in real-time across the entire IoMT and between a high range of classes which, depending on the diversity of devices deployed at scale, could result in further development of the responses. Exploration of adaptive security responses may also help strengthen the model's ability to contend with changing, emerging, or latent malicious activity toward IoMT network data.

Declaration

Data availability

https://www.kaggle.com/datasets/vigneshvenkateswaran/bot-iot

Conflicts of Interest

The author declares that there are no conflicts of interest regarding the publication of this paper.

Author Contribution

Abdullah M. Albarrak is the sole author of this work and was responsible for the conception, design, implementation, analysis, and writing of the manuscript.

Ethical Approval

This article does not contain any studies with human participants or animals performed by the author.
Not applicable.

References

[1] Al Khatib I, Shamayleh A, Ndiaye M. Healthcare and the Internet of Medical Things: Applications, trends, key challenges, and proposed resolutions. Informatics. 2024;11(3):47.

[2] Batista E, Moncusi MA, López-Aguilar P, Martínez-Ballesté A, Solanas A. Sensors for context-aware smart healthcare: A security perspective. Sensors. 2021;21(20):6886.

[3] Alzahrani FA, Ahmad M, Ansari MTJ. Towards design and development of security assessment framework for Internet of Medical Things. Appl Sci. 2022;12(16):8148.

[4] Sadhu PK, Yanambaka VP, Abdelgawad A, Yelamarthi K. Prospect of Internet of Medical Things: A review on security requirements and solutions. Sensors. 2022;22(15):5517.

[5] Naghib A, Gharehchopogh FS, Zamanifar A. A comprehensive and systematic literature review on intrusion detection systems in the Internet of Medical Things: Current status, challenges, and opportunities. Artif Intell Rev. 2025;58(4):114.

[6] Pelekoudas-Oikonomou F, et al. Blockchain-based security mechanisms for IoMT edge networks in IoMT-based healthcare monitoring systems. Sensors. 2022;22(7):2449.

[7] Alamleh H, Estremera L, Arnob SS, AlQahtani AAS. Advanced persistent threats and wireless local area network security: An in-depth exploration of attack surfaces and mitigation techniques. J Cybersecur Priv. 2025;5(2):27.

[8] Sun S, Xie Z, Yu K, Jiang B, Zheng S, Pan X. COVID-19 and healthcare system, Challenges and progression for a sustainable future. Glob Health. 2021;17(1):14.

[9] Bhushan B, Kumar A, Agarwal AK, Kumar A, Bhattacharya P, Kumar A. Towards a secure and sustainable Internet of Medical Things (IoMT): Requirements, design challenges, security techniques, and future trends. Sustainability. 2023;15(7):6177.

[10] Zehra S, et al. Machine learning-based anomaly detection in NFV: A comprehensive survey. Sensors. 2023;23(11):5340.

[11] Heidari A, Jabraeil Jamali MA. Internet of Things intrusion detection systems: A comprehensive review and future directions. Cluster Comput. 2023;26(6):3753–3780.

[12] Entezami A, Sarmadi H, Behkamal B, Mariani S. Early warning of structural damage via manifold learning-aided data clustering and non-parametric probabilistic anomaly detection. Mech Syst Signal Process. 2025;224:111984.

[13] Elsayed R, Hamada R, Hammoudeh M, Abdalla M, Elsaid SA. A hierarchical deep learning-based intrusion detection architecture for clustered Internet of Things. J Sens Actuator Netw. 2023;12(1):3.

[14] Zachos G, Essop I, Mantas G, Porfyrakis K, Ribeiro JC, Rodriguez J. An anomaly-based intrusion detection system for Internet of Medical Things networks. Electronics. 2021;10(21):2562.

[15] Alsoufi MA, et al. Anomaly-based intrusion detection systems in IoT using deep learning: A systematic literature review. Appl Sci. 2021;11(18):8383.

[16] Toldinas J, Venčkauskas A, Damaševičius R, Grigaliūnas Š, Morkevičius N, Baranauskas E. A novel approach for network intrusion detection using multistage deep learning image recognition. Electronics. 2021;10(15):1854.

[17] Nayak J, Meher SK, Souri A, Naik B, Vimal S. Extreme learning machine and Bayesian optimization-driven intelligent framework for IoMT cyber-attack detection. J Supercomput. 2022;78(13):14866–14891.

[18] Chaganti R, Mourade A, Ravi V, Vemprala N, Dua A, Bhushan B. A particle swarm optimization and deep learning approach for intrusion detection system in Internet of Medical Things. Sustainability. 2022;14(19):12828.

[19] Ullah S, et al. HDL-IDS: A hybrid deep learning architecture for intrusion detection in the Internet of Vehicles. Sensors. 2022;22(4):1340.

[20] Alalhareth M, Hong SC. An adaptive intrusion detection system in the Internet of Medical Things using fuzzy-based learning. Sensors. 2023;23(22):9247.

[21] Vishwakarma M, Kesswani N. A new two-phase intrusion detection system with naïve Bayes machine learning for data classification and elliptic envelope method for anomaly detection. Decis Anal J. 2023;7:100233.

[22] Si-Ahmed A, Al-Garadi MA, Boustia N. Survey of machine learning based intrusion detection methods for Internet of Medical Things. Appl Soft Comput. 2023;140:110227.

[23] Vijayakumar KP, Pradeep K, Balasundaram A, Prusty MR. Enhanced cyber attack detection process for Internet of Health

Things devices using deep neural network. Processes. 2023;11(4):1072.

[24] Sun Z, An G, Yang Y, Liu Y. Optimized machine learning enabled intrusion detection system for Internet of Medical Things. Franklin Open. 2024;6:100056.

[25] Talukder MA, Sharmin S, Uddin MA, Islam MM, Aryal S. MLSTL-WSN: Machine learning-based intrusion detection using SMOTETomek in WSNs. Int J Inf Secur. 2024;23(3):2139–2158.