Ibn Al-Haitham Journal for Pure and Applied Sciences
Journal homepage: jih.uobaghdad.edu.iq
PISSN: 1609-4042, EISSN: 2521-3407
IHJPAS. 2026, 39(1)

An Intrusion Detection for Internet of Medical Things Based on Deep Learning Techniques

Aisha Essa Mohammad [1*], **Amer Abdulmajeed Abdulrahman** [2]
[1,2] Computer Science Department, College of Sciences, University of Baghdad, Baghdad, Iraq
*Corresponding Author

Received: 4/June/2025
Accepted: 27/August/2025
Published: 20/January/2026.
doi.org/10.30526/39.1.4222

Abstract

The Internet of Medical Things (IoMT) has enhanced healthcare, but it is also vulnerable to cyber-attack. Reliable Intrusion Detection Systems (IDSs) are essential for data integrity and patient safety. The goal of this study is to design and evaluate deep learning IDSs in the form of DNNs and a hybrid GRU-DNN model for IoMT networks using the CICIoMT2024 dataset.

The design of a standalone DNN model and a hybrid Gated Recurrent Unit–DNN (GRU-DNN) was also applied and compared. Both were trained and tested with various classes involving 2, 6, and 19 from the CICIoMT2024 datasets. The GRU-DNN model, in 19-class classification performance, achieved satisfactory results of accuracy, precision, and recall as 98.4%, 98.6%,98.4% and 98.2% respectively, for F1-score. The DNN model achieved 93.1 % accuracy for the same task. The model outperformed other models, such as LSTM and previous DNN models.

The proposed hybrid GRU-DNN model exhibits promising results in being applicable to identifying intrusions in IoMT systems and seems to hold great promise for improving the security of real clinical networks.

Keywords: Internet of Medical Things (IoMT), Deep Neural Network (DNN), Gated Recurrent Unit (GRU), Deep Learning (DL), Intrusion Detection System (IDS).

1. Introduction

The Internet of Medical Things (IoMT) is a new application scenario that extends the Internet of Things (IoT), in which IoMT devices are deployed to provide medical services. The increasing number of connected medical devices on the Internet has significantly broadened the concept of digital healthcare services. This ubiquity has brought about significant security issues and vulnerabilities[1-3].

IoMT has revolutionized medical care, making monitoring, remote diagnosis, and personalized treatment via connected medical devices all a reality. However, the fast penetration of IoMT devices has brought about serious cyberspace concerns because such things are highly diverse in type and have scarce resources, yet are particularly critical given that medical data is involved. Attacks on IoMT systems, including Distributed Denial of Service (DDoS) attacks, spoofing, and protocol-specific attacks, will result in safety issues and data integrity. Even when securely transported over networks, data is permanently at risk of unauthorized intrusion and misuse. Conventional IDS is frequently ill-suited for IoMT, such as due to resource limitations, low latency requirements, and device diversity. These constraints substantially hamper their ability to properly detect and address advanced threats, underscoring the necessity for more evolved security solutions[4-6].

In the IoMT field, deep learning techniques are becoming popular for improving IDS's performance due to their capability of capturing complex temporal dynamics and non-linearity in high-dimensional network traffic data.

Small generic architectures such as Deep Neural Networks (DNNs) and Gated Recurrent Units (GRUs) have been shown to learn complex patterns and relationships in network traffic, as indicated by [7].

It was proposed the CICIoMT2024 dataset for a Machine Learning (ML) and DNN study[4]. Their approach succeeded in taking accurate binary classification with gene markers of 99.6% accuracy, as shown in their experimental results. The model exhibited a mediocre accuracy of 73.4% in the 6-class classification. But it performed relatively worse in a 19-class classification task with an accuracy of 72.9%.

It was presented the L2D2 model, which is based on an LSTM structure and is particularly tailored for the detection of attacks involving multiple classes in IoMT[8]. The model performed well in terms of classification on the CICIoMT2024 dataset, with 100% accuracy for binary, 98% for 6-class, and 95% for converted to the other 19-classes.

Most existing IDS studies focused on binary classification and did not address the complexity of multi-class detection in IoMT environments.

Therefore, this study aims to develop a deep learning-based multi-class intrusion detection system for IoMT using the CICIoMT2024 dataset.

This research aims to contribute to the creation of safe, sustainable, and intelligent healthcare infrastructures that are compatible with the growing popularity of digital health and patient safety.

The remainder of this paper is structured as follows: Section 2 describes the Materials and Methods; Section 3 presents the Results and Discussion; and Section 4 concludes the paper.

2. Materials and Methods

The Materials and Methods include a detailed description of the CICIoMT2024 dataset, the applied data preprocessing steps, and the deep learning models used for intrusion detection.

2.1 CICIoMT2024 Dataset

The Canadian Institute for Cyber Security created the CICIoMT2024 dataset as a comprehensive means of testing the safety of medical technology associated with the Internet of Medical Things (IoMT). It contains traffic from 40 IoMT devices: 25 real and 15 simulated, under attack of eighteen types of cyber threats against three protocols: Wi-Fi, MQTT, Bluetooth[4].

The CICIoMT2024 dataset has 45 features, covering various aspects of network traffic. This is to monitor unusual and malicious behavior, as explained by.[9].

Although the CICIoMT2024 dataset is an extensive benchmark for testing IDSs in IoMT, it does have some limitations that could affect the model's performance. For example, some classes are relatively under-represented in the distribution of attack types. This may cause learning bias and result in poor detection performance of those attacks. In addition, some of the dataset is the traffic created by virtual devices, which may not accurately represent the actual situation. These constraints need to be considered when designing and testing detection models.

2.2 Data Preprocessing

In order to form the data for training and achieve good performances in terms of accuracy, but also efficiency, before feeding it into our model, several preprocessing steps are required. The preprocessing enhances the data quality of the models as described by [10].

Preprocessing manipulations are Data Cleaning, Reshaping, Feature Selection, and Data Transformation (Encoding Categorical variables (Label encoding, One-hot encoding), Feature Scaling (Standardization), and Shuffling).

2.2.1 Data Cleaning

Date cleaning is the process of identifying, fixing, or removing errors and inconsistencies, missing data values, and duplicate records from a dataset in order to correct or improve its quality for analysis or modeling. It assures the correctness of the data before being used [11,12].

No missing values were found, and some duplicates were found. The training file has 5119 duplicate values, and the test file has 2065 removed duplicate values. This prevented overfitting by splitting off the repeated patterns, which had the potential to bias or overfit our model.

2.2.2 Reshaping

The conversion of feature vectors into a three-dimensional (3D) format (samples, time steps, features) is considered a pre-processor-specific step, so it is crucial to ensure the compatibility of its output with sequential neural structures like LSTM and GRU. This type of structural adjustment is necessary for the data pipeline to convert the input format to the expected size of the recurrent model [13-15]. The input data is converted into a 3D format (samples, time steps, features) that meets the demands of the GRU model, being a type of RNN, as explained by.[29].

In contrast, DNN, also known as feedforward or fully connected networks, expect input data in a two-dimensional (2D) format (samples, features), as noted by[30].

2.2.3 Feature Selection

It refers to the process of selecting features that are important or necessary for the model's functioning and reducing overfitting. The standard deviation of all numerical features is determined to assess their variability and importance to the model. Features with a low standard deviation near zero are considered to have little to no differentiating information and can potentially introduce noise to the learning process. This step reduced the dimensionality of the space while ensuring that the input space was composed of only attributes that had a significant effect, increasing the effectiveness and generalization capacity of the model, as noted by[16].

2.2.4 Encoding and Scaling

It refers to converting categorical labels into numerical values. This step is essential for allowing deep learning models to utilize the class label to train effectively. Data transformation consists of encoding categorical variables that contain (Label encoding, One-hot encoding), and Feature Scaling (Standardization).

- Label encoding
  The classes are converted into integers that represent machine-readable numerical values[17,18].
- One-hot encoding
  Encoded labels are transformed into binary vectors, enabling the model to perform multi-class classification through SoftMax activation in the output layer[19,20]. Labels are successfully assigned to 2, 6, and 19 classes. It was observed that their distribution was uniform. All categorical labels were converted to one-hot encoding.
- Feature Scaling (Standardization)
  To ensure that the numerical inputs are of the same scale, Z-score normalization is employed using the StandardScaler. This change altered each attribute to possess a mean of 0 and a standard deviation of 1. Standardization was crucial to prevent attributes with larger variations from disproportionately affecting the learning process and to increase the rate of convergence during model training. This preprocessing procedure had a positive effect on the overall stability and performance of the deep learning model [21,22].

2.2.4 Data Shuffling

It is the process of randomly reordering the samples in a dataset. It is not a transformation of feature values, but rather a reorganization of sample order to prevent the model from learning spurious patterns based on the input sequence. This is especially important for neural networks [23-25].

2.3 The Deep Learning Models

After collecting the data in the pre-processing stage, it went through multiple processing stages, including the selection of essential features, and then the number of features was reduced to 44. The information is now prepared to be evaluated for the classification stage. In this phase, several scenarios will be employed for different DL algorithms, including: The hybrid model of GRU-DNN and DNN was performed in three different types of classification (2-class, 6-class, and 19-class) as experiments to maximize the accuracy and efficiency.

2.3.1 The DNN Model

The Deep Neural Network (DNN) is a robust design that has multiple layers of hidden information that facilitate the acquisition of complex features. In the context of the IoMT, which has a data communication that is typically high-dimensional, diverse, and vulnerable to various types of cyberattacks, DNNs have a beneficial effect on intrusion detection. Through their complex, hierarchical design, DNNs can learn intricate, complex patterns in the behavior of networks. This distinction is necessary to differentiate between legitimate and criminal traffic [26,27].

The DNN architecture consists of two fully connected layers, both of which are extensive, followed by an output layer with the same number of units as the number of classes related to it (i.e., num_classes).

The first and the second dense layers consist of 64 and 32 units respectively, that is activated by ReLU function. This function is widely used due to its non-linearity and power in deep learning tasks. We apply dropout regularization after each dense layer with a dropout rate of 0.3, i.e., removing 30% of the neurons randomly during training in order to prevent overfitting. SoftMax activation function used at the output layer for multi-class classification. This function outputs a distribution of the likelihood over target classes.

The Adam optimizer is chosen since it can enable faster computation time and the adaptive learning rate feature, as reported by[28].

The initial value of the learning rate was set at 0.0001 to make our training process's results stable. The loss function being applied is Categorical Cross-Entropy, common in problems with multiple classes.

The model was trained for 50 epochs with a batch size of 64. Early stopping with a patience of 5 epochs was applied to avoid overfitting; it stopped training when no decrease in validation loss was achieved for five consecutive epochs.

We chose this setting according to the empirical guideline so that a trade-off can be reached among model complexity, generalization ability, and computational efficiency.

2.3.2 The Proposed Hybrid Model (GRU-DNN)

There is a previously proposed design, the Grated Recurrent Unit–Deep Neural Network Hybrid Model (GRU-DNN), which is intended to meet the dynamic and real-time demands of cybersecurity in the IoMT. This architecture is an amalgamation of GRUs, capturing the strength in temporal processing and DNN to leverage the power of deep learning. It allows us to model complex environments and highly automated traffic flows.

The collaboration not only contributes to the generalization and expansibility of medical networks, but also achieves a remarkable effect in a medical network that contains various data structures and massive data.

The model workflow is schematically outlined in the following:

Load the training and testing files from the dataset. The dataset was split into an 80% set for training and a 20% test set according to the given structure. Make sure the data are shuffled to have some randomness and no bias. Remove irrelevant and redundant features, e.g., 'Drate'. Drop the missing and duplicate values. Map the original attack labels to one of three class categories in classification: The binary, indicating Benign vs. Attack; The six-class, which represents (DDoS, DoS, etc., and MQTT types respectively...), and Full 19-class classification. Preprocess your

data: Encode the labels using LabelEncoder. Convert to_categorical one-hot with Keras. Normalize features using StandardScaler. Reshape input for GRU: (samples, 1, features).

The hybrid GRU-DNN model is structured in a sequential fashion. The GRU layer firstly captures the temporal features in the input sequences. Normalization and regularization of these features are done using Batch Normalization and Dropout. These are then input into a stack of fully connected layers to generate high-level features. The last output layer uses SoftMax to calculate the attack class. This architecture is also drawing on the complementary abilities of RNN and deep feedforward learning. The architecture of the Layers used in the hybrid model is depicted in **Figure 1**.

The workflow of the proposed method is depicted in **Figure 2**, which shows four main components of the process: dataset, data preprocessing step, the model used, and evaluation.

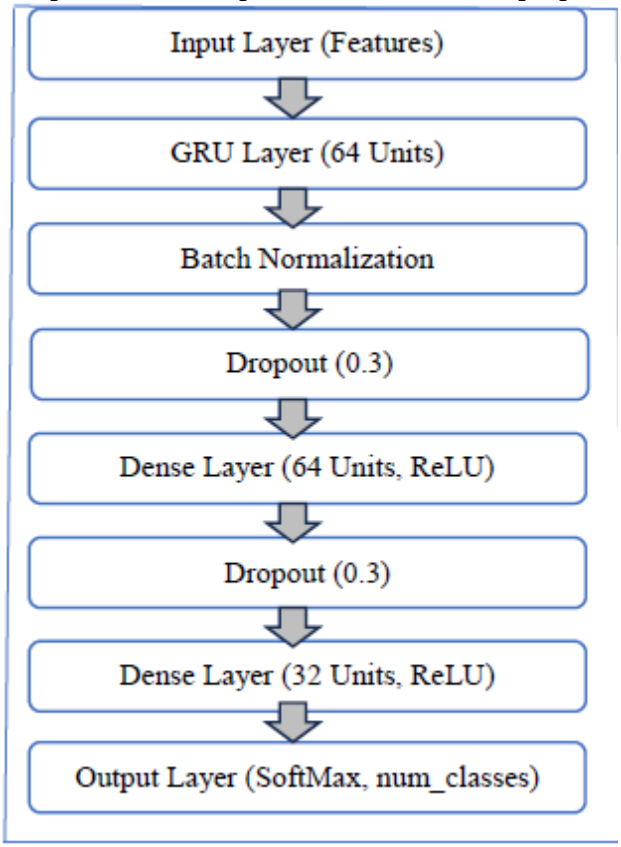

Figure 1. Structure of the layers in the hybrid model (GRU-DNN)

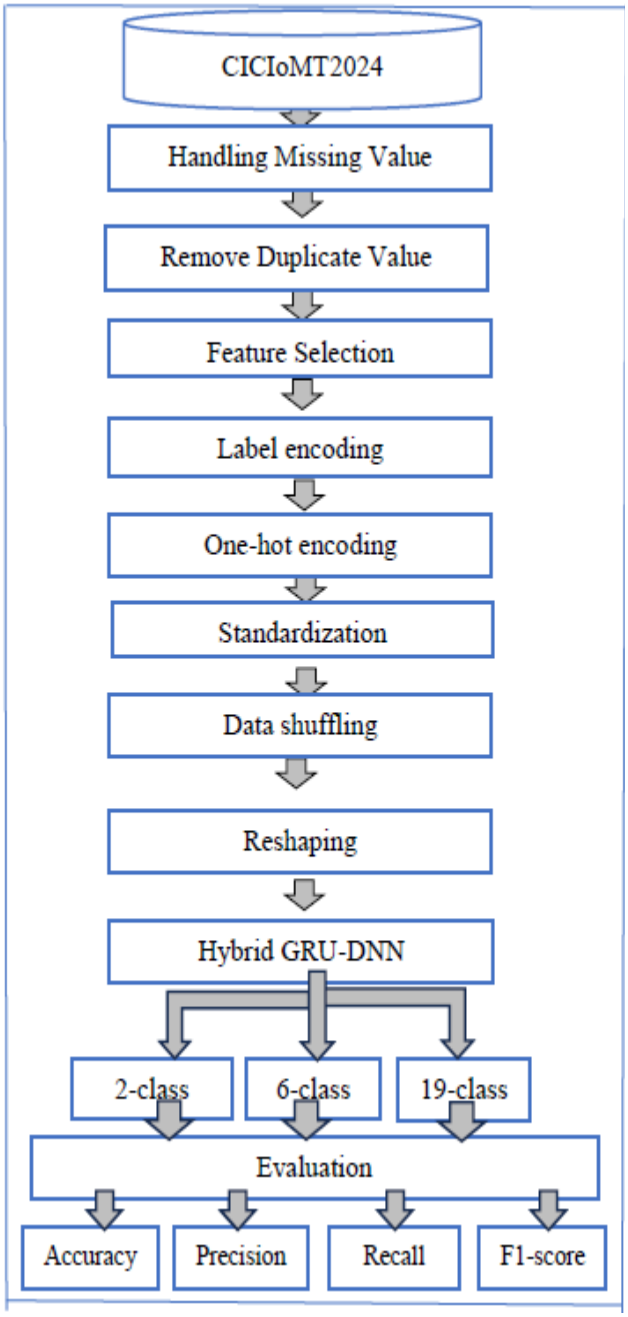

Figure 2. Diagram of the proposed hybrid model (GRU-DNN)

Compile the model: Loss: categorical_crossentropy. Optimizer: Adam, learning rate = 0.0001. 20% of the training set was additionally used to serve as a validation set in order to control the model's generalized with early stopping. Early Stopping with patience=5. Train for at most 50 epochs. Use batch size = 64. did not use cross-validation to save time on training and computation.

Test the model to find accuracy, precision, recall, and F1-score. Generate a classification report. Plot the confusion matrix and training history (accuracy/loss) plot.

The parameters and training configurations for the hybrid and independent models are summarized in **Table 1**.

Table 1. Parameters and training setup used in the hybrid and standalone models

Parameters	Optimizer	Learning Rate	Loss Function	Epochs	Batch Size	Early Stopping Patience
value	Adam	0.0001	Categorical Cross-entropy	50	50	5

3. Results and Discussion

The findings reveal the effect of data preprocessing, deep learning model performances, and the adopted evaluation metric. A comparison with the related work on the CICIoMT2024 dataset is presented to demonstrate the potency of the proposed method.

3.1 Deep Learning Model's Results

To evaluate the performance of the framework for different classification scenarios (binary, multi-class, and Complex Multi-Class) based on selected model parameters, a deep learning-based classification technique was used in this research.

The CICIoMT2024 dataset is capable of handling different levels of attack classification. The choice of 2, 6, and 19-class settings is beneficiary from the hierarchical nature of attack categorization in the CICIoMT2024 dataset that reflects security risk in real-world IoMT deployments.

For the classification levels, accuracy and loss were monitored across epochs as well as the confusion matrix. The results are all the metrics, including accuracy, precision, recall, and F1-score.

3.1.1 DNN Model Results

- Binary classification (2-class)

  The performance of the DNN model in binary classification involves two classes: Attack and Benign in the IoMT environment.

Figure 3 below depicts the training and validation performance of the DNN model in the 2-class classification task. The left plot of the graph shows the evolution of accuracy, while the right plot exhibits the corresponding loss graphs.

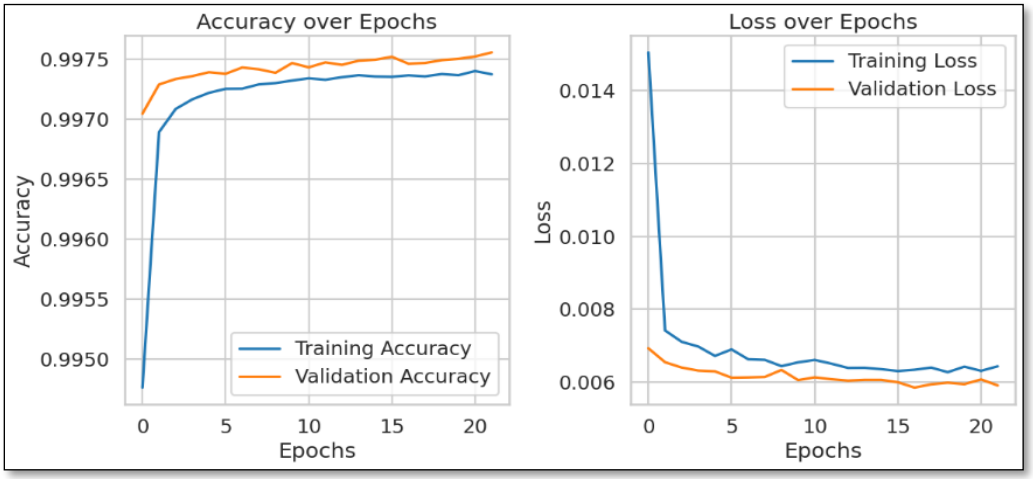

Figure 3. Accuracy and Loss over epochs for the DNN Model in 2-class

The confusion matrix in **Figure 4** illustrates the binary classification results. Diagonal values (1,571,981 and 34,151) represent correctly classified samples, while off-diagonal values indicate misclassifications.

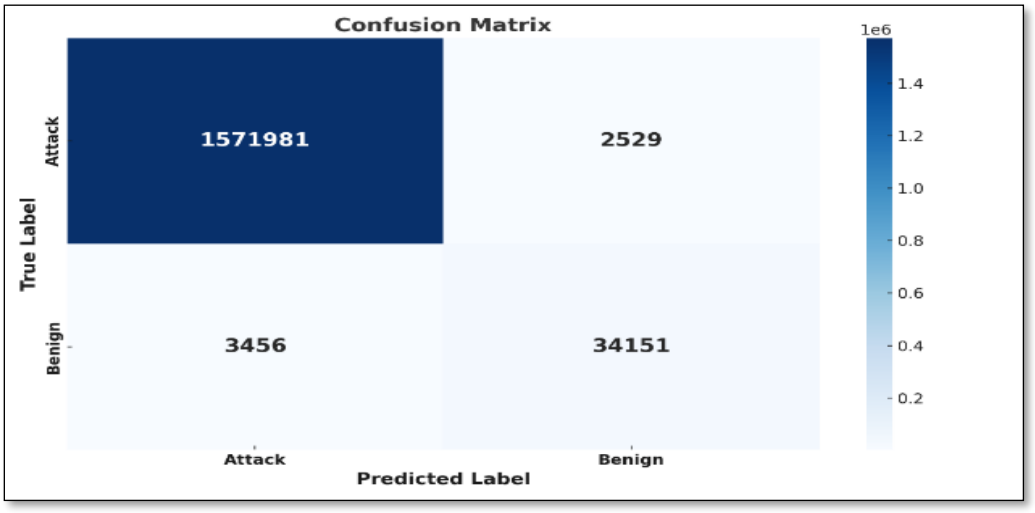

Figure 4. Confusion Matrix for the DNN Model in 2-class

To evaluate the performance of the DNN model and the proposed hybrid model (GRU-DNN) in all classes, we used standard classification metrics, including:

- Accuracy
  Accuracy measures the proportion of correctly classified instances among all instances.
  $$\text{Accuracy} = \frac{TP+TN}{TP+TN+FP+FN} \quad (1)$$
  - TP (True Positives): Correctly predicted positive classes.
  - TN (True Negatives): Correctly predicted negative classes.
  - FP (False Positives): Incorrectly predicted as positive.
  - FN (False Negatives): Incorrectly predicted as negative.
- Precision
  Precision measures how many of the predicted positive classes were actually correct.
  $$\text{Precision} = \frac{TP}{TP+FP} \quad (2)$$
- Recall (Sensitivity)
  Recall measures how many of the actual positive instances were correctly predicted.
  $$\text{Recall} = \frac{TP}{TP+FN} \quad (3)$$
- F1-Score
  Harmonic mean of Precision and Recall, balancing both metrics.
  $$\text{F1-Score} = 2 \times \frac{Pre \times Rec}{Pre \times Rec} \quad (4)$$
  - Precision (Pre).
  - Recall (Rec).

These metrics allow for a comprehensive assessment of the model's ability to classify both normal and attack classes correctly.

Table 2 below shows the final results, confirming the strong performance.

Table 2. Final results for the DNN Model in 2-class

Accuracy	Precision	Recall	F1-Score
99.6%	99.6%	99.6%	99.6%

- Multi-Classification (6-Class)
  It deals with the 6-class task of classifying things considered benign and those that are attacks. These attacks are divided into five main classes: DDoS, DoS, MQTT, RECON, and SPOOFING.

Figure 5 consists of two plots displaying the training and validation performance of a deep learning model. The left plot demonstrates a consistent increase in both the training and validation accuracy over the epochs. The right plot shows a consistent decrease in both training and validation loss.

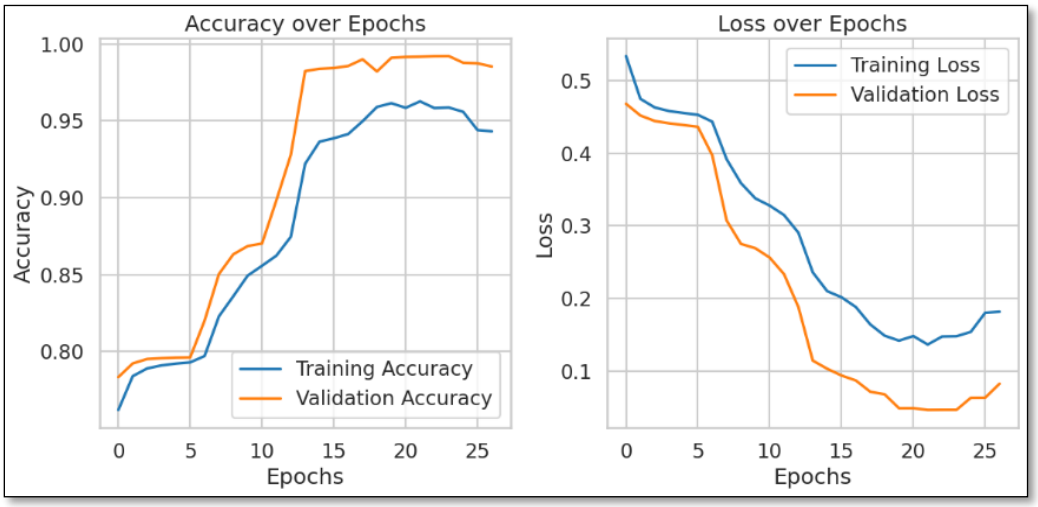

Figure 5. Accuracy and Loss over epochs for the DNN Model in 6-class

The confusion matrix in **Figure 6** for the 6-class classification shows strong performance, with high values along the diagonal indicating correct predictions. Misclassifications are relatively low, suggesting the model effectively distinguishes between most attack types and benign traffic.

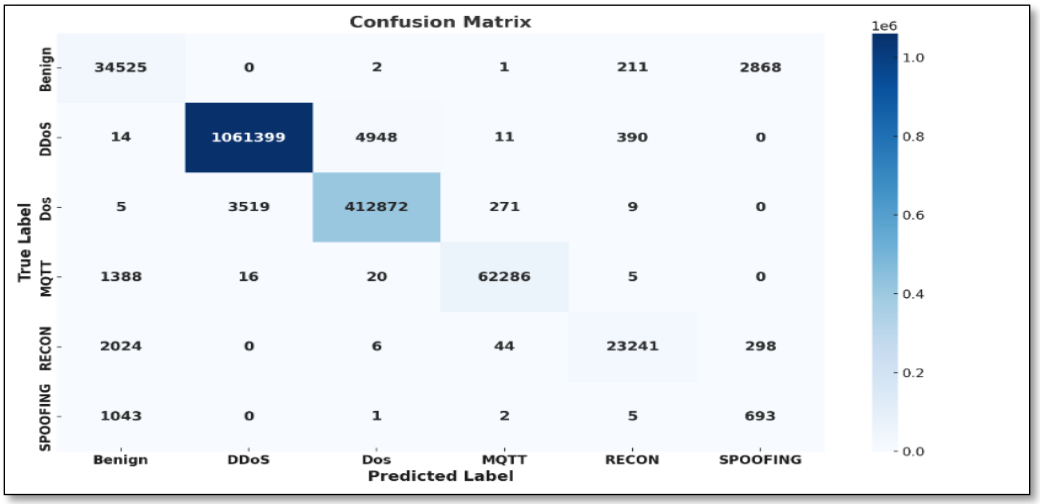

Figure 6. Confusion Matrix for the DNN Model in 6-class

The results in **Table 3** exhibit the impressive performance of the DNN model in the 6-class categorization task.

Table 3. Final results for the DNN Model in 6-class

Accuracy	Precision	Recall	F1-Score
98.9%	99.1%	98.9%	99.0%

- **Complex multi-classification (19-Class)**
  The model is capable of distinguishing between different classes of attack traffic, which includes 18 types of attacks and normal behavior.

The training and validation performance of the model for 35 epochs is illustrated in **Figure 7**. On the left: the accuracy curve is monotonically increasing, while the validation model overtakes the training model after 15 epochs. The maximum achieved points for the accuracy were 93%.
This indicates a significant capacity for generalization. On the right, the loss curve exhibits a consistent decrease, with the validation loss being lower than the training loss, which is approximately 0.15.

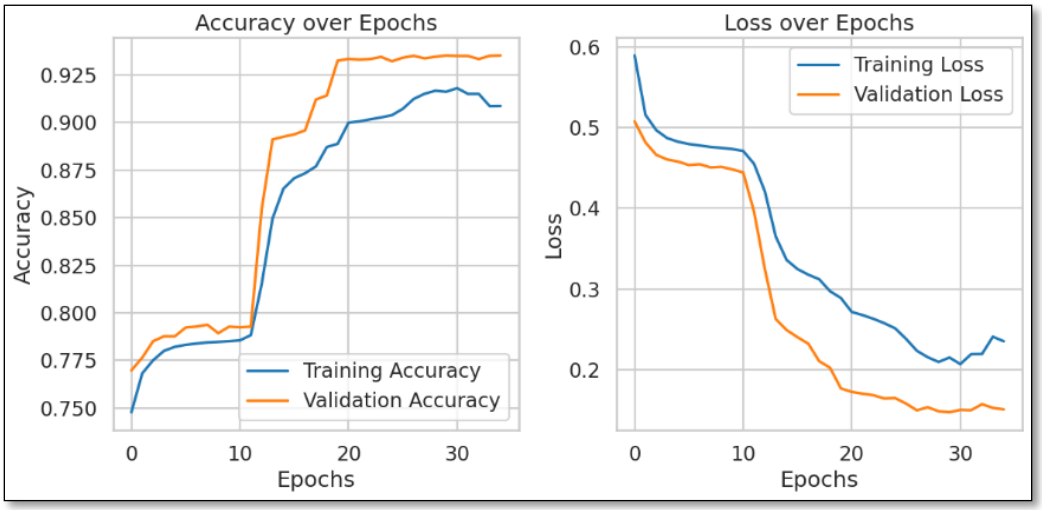

Figure 7. Accuracy and Loss over epochs for the DNN Model in 19-class

The confusion matrix in **Figure 8** demonstrates how well the model can differentiate between different types of attack traffic and benign. The Diagonal values represent instances that are correctly classified for each class, whereas the Off-diagonal values represent instances that are misclassified.

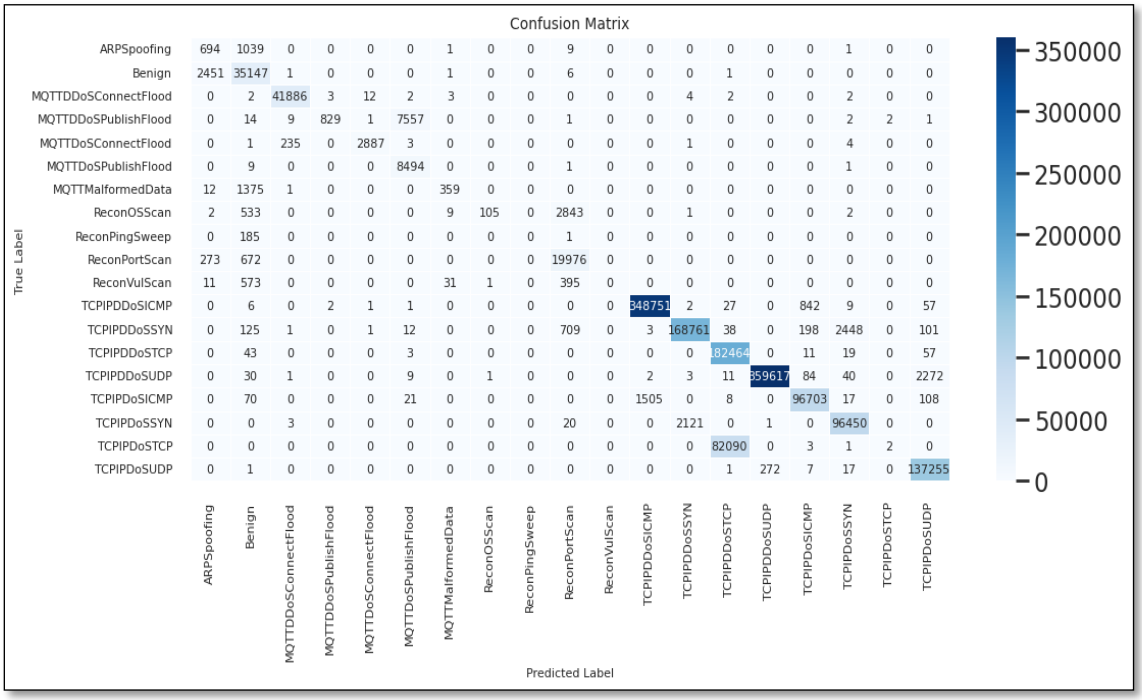

Figure 8. Confusion Matrix for the DNN Model in 19-class

The final results in **Table 4** show the overall activity of the DNN model in the 19-class categorization task.

Table 4. Final results for the DNN Model in 19-class

Accuracy	Precision	Recall	F1-Score
93.1%	92.4%	93.1%	90.8%

3.1.2 The Proposed Hybrid Model (GRU-DNN) Result

- Binary classification (2-class)
  The hybrid GRU-DNN model is employed in the binary classification task (Attack or Benign).

Figure 9 shows the training and validation performance of the hybrid GRU-DNN model for the binary classification task (Attack and Benign). The graph on the left shows a consistent incremental increase in both training and validation accuracy over 16 epochs, with both graphs converging around 99.6%. This indicates a highly accurate process that is not overfit. The graph on the right illustrates the loss of training and validation over the same epochs.

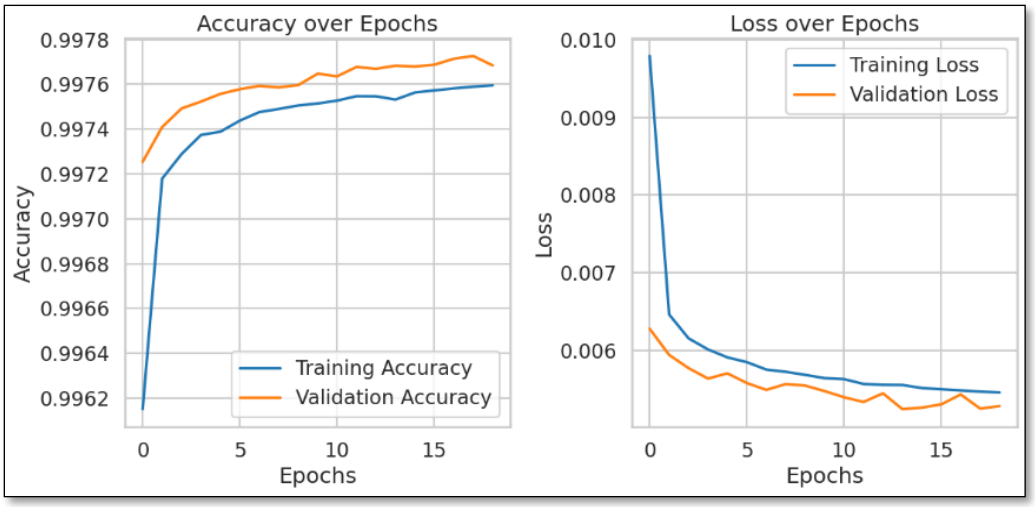

Figure 9. Accuracy and Loss over epochs for the Hybrid Model (GRU-DNN) in 2-class

The confusion matrix in **Figure 10** for the hybrid GRU-DNN model in the binary classification task (Attack vs. Benign) demonstrates high accuracy. The model accurately classified 1,572,343 attacks and only 33,652 benign samples with a tiny number of misclassifications. These values demonstrate the high sensitivity and specificity of the model, proving that it can successfully detect cyber-attacks without many false positives and negatives.

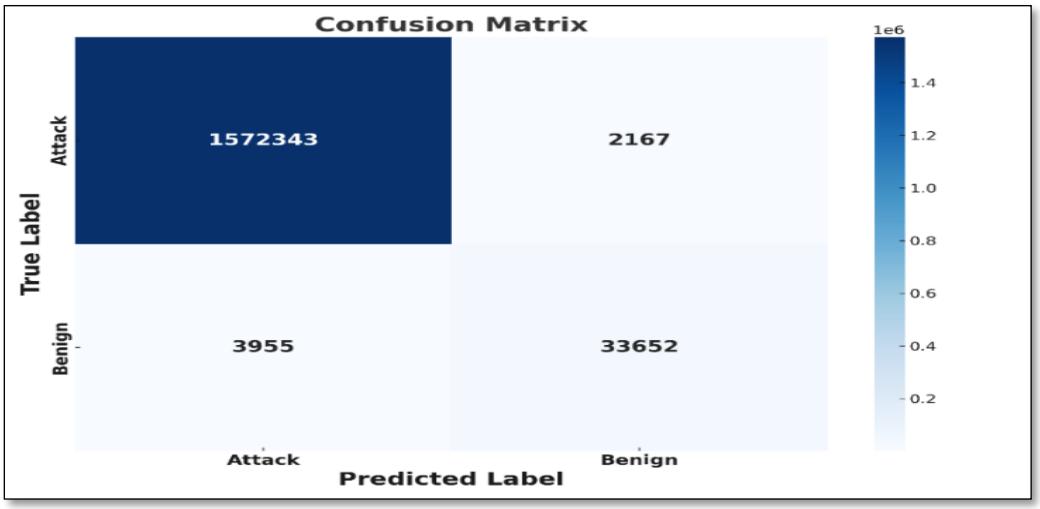

Figure 10. Confusion Matrix for the Hybrid Model (GRU-DNN) in 2-class

Table 5 summarizes the overall metrics: Accuracy, Precision, Recall, and F1-Score, which are all at 99.6%. These high scores are indicative of the effectiveness, sensitivity, and precision of the model, which makes it very reliable with respect to the detection of intrusions in IoMT networks.

Table 5. Final results for the Hybrid Model (GRU-DNN) in 2-class

Accuracy	Precision	Recall	F1-Score
99.6%	99.6%	99.6%	99.6%

- Multi-Classification (6-class)
  The hybrid GRU-DNN model is used for 6-class classification. These are referred to as benign and attacks. Those attacks currently fall under five major categories: DDoS, DoS, MQTT, RECON, and SPOOFING.

The training and validation performance of the proposed hybrid GRU–DNN model for the 6-class intrusion detection problem is depicted in **Figure 11** across 40 epochs. The left figure shows validation and training accuracy, while the right figure shows the loss curves.

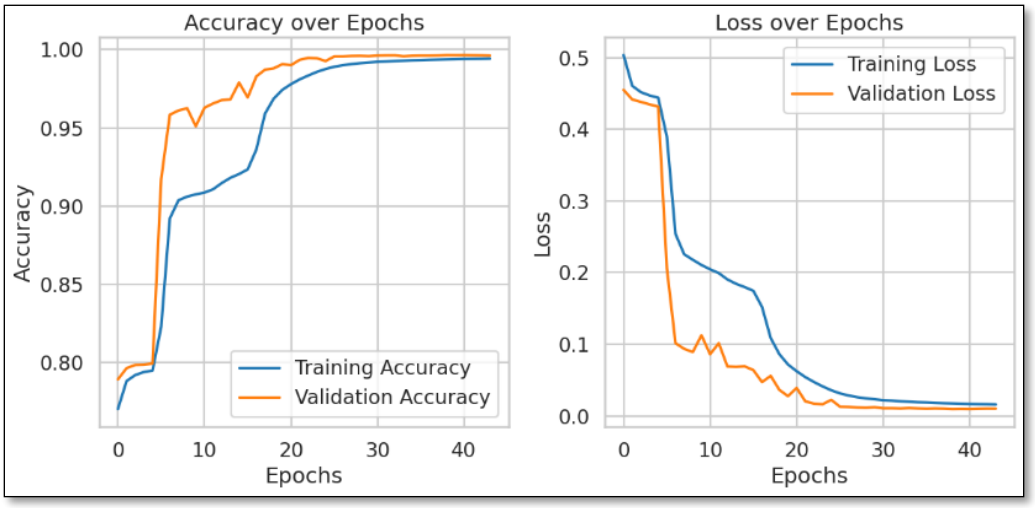

Figure 11. Accuracy and Loss over epochs for the Hybrid Model (GRU-DNN) in 6-class

The confusion matrix of the test results on 6-class for GRU-DNN model is shown in **Figure 12**, where the major classes DDoS, DoS, MQTT, and Benign are highly accurate. Small misclassifications were detected, especially in RECON and SPOOFING, which can be attributed to class imbalance and feature commonality.

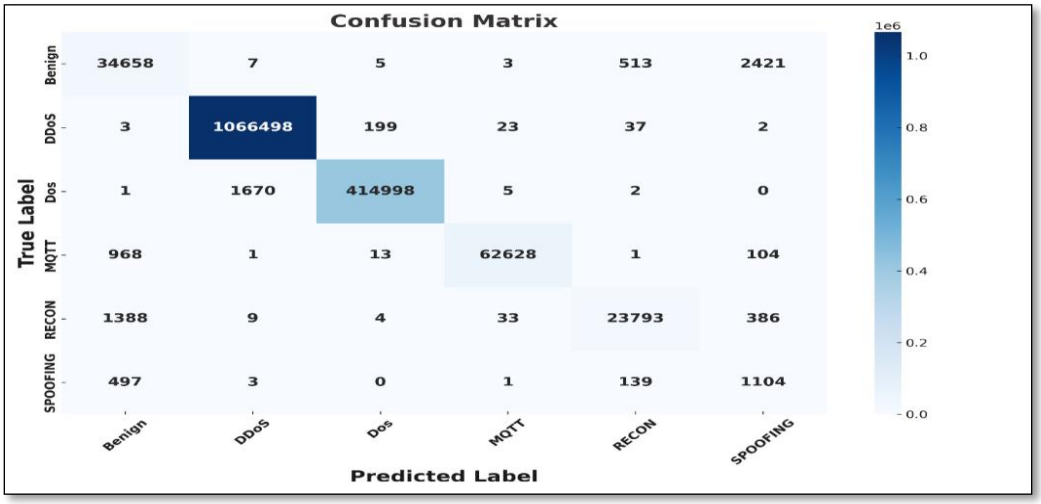

Figure 12. Confusion Matrix for the Hybrid Model (GRU-DNN) in 6-class

Table 6 shows that the GRU-DNN model performs well in detecting different types of attacks in the IoMT networks.

Table 6. Final results for the Hybrid Model (GRU-DNN) in 6-class

Accuracy	Precision	Recall	F1-Score
99.5%	99.6%	99.5%	99.5%

- Complex multi-classification (19-class)
  The proposed hybrid deep learning model using both Gated Recurrent Units (GRU) and Deep Neural Networks (DNN) is adopted on a 19-class classification task using the CICIoMT2024 dataset.

The performance of the model on identifying Benign traffic is impressive, where 33,932 samples have been identified. The model clearly has a good identifying capability against different MQTTDDoS attacks, such as MQTTDDoSConnectFlood and MQTTDDoSPublishFlood. In addition, despite the same temporal trend as well as the identical payload, both attacks tend to be more misclassified than in the GRU-DNN model. It shows that the model can learn temporal patterns.

Classes such as TCPIPDDoSICMP, TCPIPDDoSSYN, TCPIPDDoSTCP and TCPIPDDoSUDP also achieved a very high precision and recall.

For example, the model accurately predicted 349,286 samples of TCPIPDDoSICMP. This suggests that the hybrid model has the capacity to deal with packet-based features and differentiate subtle differences in the way TCP is being attacked.

Some difficulty is apparent in classes with lower representation, such as ReconPortScan and ReconOSScan, which have a higher rate of misclassification. This may be caused by an imbalanced class or by shared spaces for overlapping recognition attacks.

Figure 13 shows the training and validation performance of the hybrid GRU-DNN model across 50 epochs; the performance is evaluated on a 19-class classification task.

The graph on the left shows the increase in accuracy, followed by a flat line. The validation accuracy is 98.4%, while the training accuracy is 94%, indicating that the model has acquired the task without excessive training.

The graph on the right, i.e., the loss function, supplements this observation. Both the training and validation costs decrease significantly over time, which is particularly true for the early epochs. The validation loss is minimal and consistent at around zero, which indicates that the model is effective at generalizing to unobserved data.

These curves in total prove the robustness and effectiveness of the GRU-DNN mixed model on multi-class types for intrusion detection under an IoMT environment.

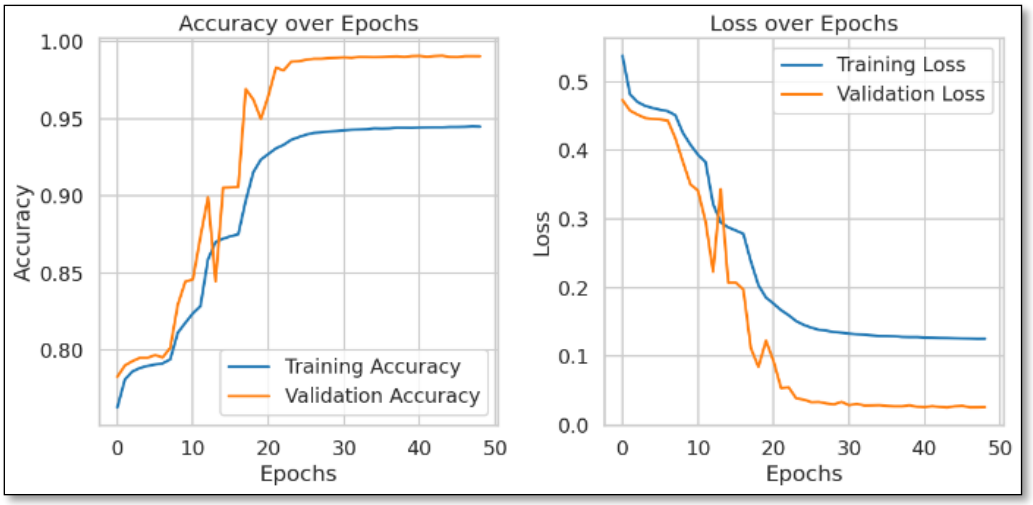

Figure 13. Accuracy and Loss over epochs for the Hybrid Model (GRU-DNN) in 19-class

Strong classification results are evident from the confusion matrix, Figure 14, for the GRU-DNN hybrid model on the 19-class. The effectiveness of this model is emphasized by correctly identifying Benign traffic and being able to differentiate with same malicious type fully due to temporal behaviour patterns. It also shows high precision and recall on the TCPIPDDoS attacks, which indicates that it can detect classes such as ReconPortScan, ReconOSScan, and present gross misclassification rates because of class bias or feature overlap. Conclusively, the results reflect the model's robustness and its appropriateness for intelligent intrusion detection in IoMT environments.

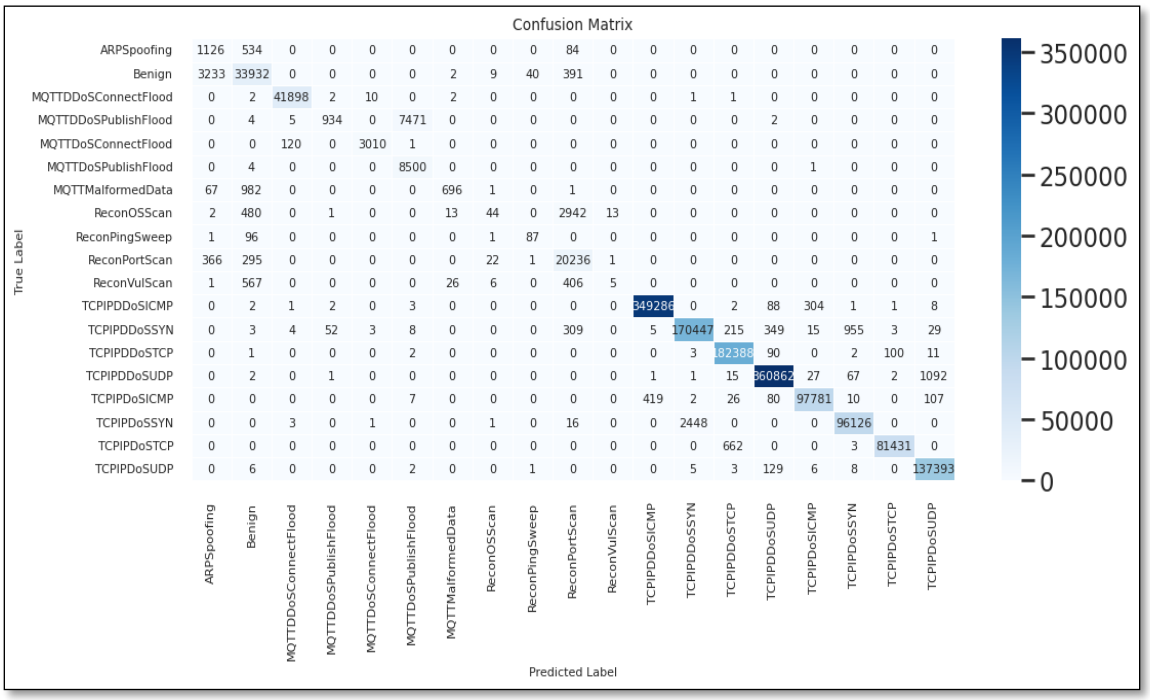

Figure 14. Confusion Matrix for the Hybrid Model (GRU-DNN) in 19-class

The 19-class classification task final metrics of the hybrid GRU-DNN model are listed in **Table 7**. These results demonstrate that the model can discriminate between legitimate and malicious traffic with a high recall rate or precision. The reported metrics, consistent across the subsets, are a testament to the applicability of the model in detecting intrusions in the IoMT environment.

Table 7. Final results for the Hybrid Model (GRU-DNN) in 19-class

Accuracy	Precision	Recall	F1-Score
98.4%	98.6%	98.4%	98.2%

3.2 Comparison with Related Work

Table 8 demonstrates the proposed GRU-DNN hybrid model with other models that were reported earlier using the CICIoMT2024 dataset. The comparison is performed on different classification duties, i.e., 2-class, 6-class, and 19-class, to prove the range of model consistency and its superiority in precision, recall, accuracy, and F1-score. Hyperparameters as well as training conditions were consistent over all of our models to make a fair comparison.

The proposed GRU-DNN hybrid model demonstrates outstanding performance, surpassing other models across all scenarios.

Table 8. Comparison with Related Work on the CICIoMT2024 dataset

Model	No. of class	Accuracy	Precision	Recall	F1-Score
The DNN model	2	99.6%	99.6%	99.6%	99.6%
	6	98.9%	99.1%	98.9%	99.0%
	19	93.1%	92.4%	93.1%	90.8%
The proposed Hybrid Model (GRU-DNN)	2	99.6%	99.6%	99.6%	99.6%
	6	99.5%	99.6%	99.5%	99.5%
	19	98.4%	98.6%	98.4%	98.2%
DNN[4]	2	99.6%	95.6%	94.8%	95.2%
	6	73.4%	72.5%	69.3%	66.5%
	19	72.9%	64.9%	55.3%	52.2%
LSTM[8]	2	100%	100%	100%	100%
	6	98.0%	98.0%	98.0%	98.0%
	19	95.0%	96.0%	95.0%	95.0%

4. Conclusion

In this study, we developed a hybrid GRU-DNN-based IDS for multiclass intrusion detection in IoMT settings using the CICIoMT2024 dataset. The proposed approach presented a clear gain over baseline methods, in particular when dealing with difficult classification tasks. The design of the model demonstrates how deep learning's hybrids can aid medical networks in accommodating their evolving security requirements while remaining accurate. The GRU-DNN technique is also promising for securing compliant IoMT systems. The main novelty of our study is that we are the first one to employ a GRU-DNN ensemble model on the CICIoMT2024 dataset for multi-class classification (2, 6, and 19 classes), and comparisons are only made to two existing works (DNN it was using by[4] and [8] with LSTM) that used the same class configurations. Despite the good performance, there are some limitations in the proposed GRU-DNN model, especially its time-consuming training process for being complicated.

Possible future works could include further generalizing the model to new threats, considering the computational burden for real-time practical applications, and incorporating it into a real-world healthcare domain.

In summary, the GRU-DNN-based IDSs show a potential for secure and robust deployment in regulated healthcare environments, providing an effective method for safeguarding sensitive medical infrastructures.

Acknowledgment

I want to express my sincere gratitude to my supervisor, Associate Professor Dr. Amer Abdulmajeed Abdulrahman, for his guidance and assistance in completing this research. I am also grateful to the staff of the Department of Computer Science, Faculty of Science, University of Baghdad, for their assistance and cooperation. Finally, I extend my sincere thanks to everyone who supported me throughout this project.

Conflict of Interest

There are no conflicts of interest to disclose.

Funding

No funding.

References

1. Saleem. A. D.; Abdulrahman. A. A. Attacks detection in Internet of Things using machine learning techniques: a review. J. Appl. Eng. Technol. Sci. 2024, 6(1), 684–703, doi:/10.37385/jaets.v6i1.4878.
2. Mathkor. D. M.; Mathkor. N.; Bassfar. Z.; Bantun. F.; Slama. P.; Ahmad. F.; Haque. S. Multirole of the internet of medical things (IoMT) in biomedical systems for managing smart healthcare systems. J. Infect. Public. Health. 2024, 17(4), 559–72, doi:/10.1016/j.jiph.2024.01.013.

3. Thamilarasu. G.; Odesile. A.; Hoang. A. An intrusion detection system for the Internet of Medical Things. IEEE. Access. 2020, 23(9), 181560–76, doi:/10.1109/ACCESS.2020.3026260.
4. Dadkhah. S.; Neto. E. C.; Ferreira. R.; Molokwu. R.C.; Sadeghi. S.; Ghorbani. A. CICIoMT2024: A benchmark dataset for multi-protocol security assessment in IoMT. Internet. Of. Things. 2024, Feb 16, doi:/10.1016/j.iot.2024.101351.
5. Abdualrahman. A. A.; Ibrahem. M. K. Intrusion detection system using data stream classification. Iraqi. J. Sci. 2021, 30(1), 319–28, doi:/10.24996/ijs.2021.62.1.30.
6. Naghib. A.; Gharehchopogh. F. S.; Zamanifar. A. A comprehensive and systematic literature review on intrusion detection systems in the Internet of Medical Things. Artif. Intell. Rev. 2025, 58(4), 1–88, doi:/10.1007/s10462-024-11101-w.
7. Gheni. H. Q.; Al-Yaseen. W.L. Using CICIoMT2024 Dataset for an Improved Intrusion Detection System. Int. Conf. Data. Analyt. Manag. 2024, 285–301, doi:/10.1007/978-981-96-3381-4_24.
8. Akar. G.; Sahmoud. S.; Onat. M.; Cavusoglu. Ü.; Malondo. E. L2D2: A Novel LSTM Model for Multi-Class Intrusion Detection Systems in the Era of IoMT. IEEE. Access. 2025, doi:/10.1109/ACCESS.2025.3526883.
9. Lipsa. S.; Dash. R. K.; Ivković. N. An interpretable dimensional reduction technique with an explainable model for detecting attacks in Internet of Medical Things devices. Sci. Rep. 2025, 15(1), 8718, doi:/10.1038/s41598-025-93404-8.
10. Gadhar. P. M.; Kadrolli. S.S.; Malathi. S. Y. Predictive modelling of cardiovascular diseases using machine learning. Recent. Trends. Healthc. Innov. 2025, 204–10. https://www.taylorfrancis.com/chapters/edit/10.1201/9781003501367-24/predictive-modelling-cardiovascular-diseases-using-machine-learning-pavitra-gadhar-sangamesh-kadrolli-malathi.
11. Karrar. A. E. The effect of using data pre-processing by imputations in handling missing values. Indones. J. Electr. Eng. Inform. 2022, 10(2), 375–84, doi:/10.52549/ijeei.v10i2.3730.
12. Swetha. B.; Rani. K. U. AW K-NN: An Adaptive Weighted K-Nearest Neighbour Framework for Handling Missing Data. Int. J. Intell. Eng. Syst. 2025, 18(3), doi:/10.22266/ijies2025.0430.43.
13. Bouktif. S.; Fiaz. A.; Ouni. A.; Serhani. M. A. Multi-sequence LSTM-RNN deep learning and metaheuristics for electric load forecasting. Energies. 2020, Jan 13, 13(2), 391, doi:/10.3390/en13020391.
14. Azyus. A. F.; Wijaya. S. K.; Kurniawan. B. Remaining Useful Life Prediction of Turbofan Engines Using CNN-GRU. J. Mech. Civ. Ind. Eng. 2025, 6(1), 19–27, doi:/10.32996/jmcie.2025.6.1.3.
15. Habieeb. R.; Kabeel. A. E.; Abdelsalam. M. M. Leveraging AI to Enhance Water Recovery and Salt Rejection in Hybrid Reverse Osmosis Desalination Plants. Water. Conserv. Sci. Eng. 2025, 10(1), 16, doi:/10.1007/s41101-024-00330-3.
16. Peng. Wu; Zhang. Q.; Wang. G.; Yang. F.; Xue. F. Dynamic feature selection combining standard deviation and interaction information. Int. J. Mach. Learn. Cybern. 2023, 14(4), 1407–26, doi:/10.1007/s13042-022-01706-4.
17. Gatea MJ; Hameed SM. An Internet of Things botnet detection model using regression analysis and linear discrimination analysis. Iraqi J Sci. 2022;63(10):4534–46. https://doi.org/10.24996/ijs.2022.63.10.36
18. Sharma. S.; Sharma. R.; Sharma. S.; Bhavsar. A.; Dutt. V. DIAEM-L: Diabetes Ensemble Analytics Model for Non-Invasive Screening in Rural Populations. https://www.researchgate.net/publication/392193088.
19. Samuels. J. I. One-hot encoding and two-hot encoding: an introduction. Preprint. 2024, 10, doi:/10.13140/RG.2.2.21459.76327.
20. Liang. Y.; Li. M. A deep learning model for prediction of lysine crotonylation sites. Sci. Rep. 2025, 15(1), 1–2, doi:/10.1038/s41598-025-04058-5.
21. De. Amorim. L. B.; Cavalcanti. G. D.; Cruz. R. M. The choice of scaling technique matters for classification performance. Appl. Soft. Comput. 2023, 133, 109924. doi:/10.1016/j.asoc.2022.109924.
22. Chen. D.; Cheng. T.; Hong. S.; Gao. W.; Lu. Z.; Yang. L. Unsupervised Domain Adaptation for Cross-domain Remote Sensing Object Detection. Preprint, doi:/10.21203/rs.3.rs-6642304/v1.
23. Zhang. Z. Enhancing Distributed Machine Learning through Data Shuffling. ITM. Web. Conf. 2025, 73, 03018, doi:/10.1051/itmconf/20257303018.
24. Chen. F.; Xiang. L.; Sun. H.; Cheng. H. V.; Shen. K. Shuffling for Semantic Secrecy. IEEE. Trans. Inf. Forensics. Secur. 2025, doi:/10.1109/TIFS.2025.3573192.

25. Wei. H.; Gao. L.; Qi. L.; Yuan. S. A Novel Deep Learning Model for Road Obstacles Detection. SSRN, https://ssrn.com/abstract=5269355.
26. Mageed. I. A. Surpassing Beyond Boundaries: Open Mathematical Challenges in AI-Driven Robot Control. Preprint. 2025, doi:10.20944/preprints202505.2456.v1.
27. Wilk-Jakubowski M.; Pontikakis A.; Tryfonas T. Machine learning and neural networks for phishing detection. *Electronics*. 2025, 14(18), 3744. https://doi.org/10.3390/electronics14183744
28. Reyad M, Sarhan AM, Arafa M. A modified Adam algorithm for deep neural network optimization. Neural Comput Appl. 2023;35(36):25481–25494. https://doi.org/10.1007/s00521-023-08568-z.
29. Xu. S.; Wang. T.; Ding. Z.; Wang. Y.; Wan. T.; Xu. D. Estimation of lower limb torque: a novel hybrid method. Peer. J. Comput. Sci. 2025, 11, e2888, doi:/10.7717/peerj-cs.2888.
30. Kanchi. R. S.; Melanson. B.; Somasekharan. N.; Pan. S.; He. S. UniFoil: A Universal Dataset of Airfoils. arXiv. Prepr. 2025, doi.org/10.48550/arXiv.2505.21124.