

Article

A Mutual Information-Driven Hybrid 1D-CNN and XGBoost Framework for Network Intrusion Detection in IoT

Mohammed R. Jasim

1. Jabir Ibn Hayyan University for Medical and Pharmaceutical Sciences

* Correspondence: Mohammed.r.alyasiri@jmu.edu.iq

Abstract: There are a growing number of Internet of Things (IoT) interconnected systems in critical infrastructures that have made these systems vulnerable to increasingly advanced and diverse cyberattacks. However, typical intrusions systems face serious obstacles in operation due to the high dimensional telemetry data, inadequately extracting those non-linear structural patterns, and the inherent class imbalance in network traffic. To overcome these drawbacks, this paper presents an intelligent hybrid model based on Mutual Information (MI), One-Dimensional Convolutional Neural Network (1D-CNN) and eXtreme Gradient Boosting (XGBoost). First, redundant features are removed from the proposed pipeline by using an MI metric and the most discriminative flow attributes are selected by using an elbow heuristic. Later, an optimized 1D-CNN feature extractor is used to obtain non-linear representations, which are mapped to a compact deep feature manifold. Finally, a gradient boosted decision tree (GBDT) classifier is adapted to minimize the class imbalance and achieved a good separation between the benign and malicious flows. The proposed framework achieved an overall accuracy of 98.91%, an Area under the ROC Curve (AUC) of 0.9986, an intrusion class precision of 99.35%, an intrusion class recall of 99.27%, and an F1F_1-score of 99.31% by doing extensive empirical evaluations on the latest ToN-IoT benchmark dataset containing 37,979 independent test flow instances. Comparative analysis validates the effectiveness and reliability of this framework with the basic machine learning and deep learning approaches, giving a proper balance between discrimination and avoiding the majority class bias.

Citation: Jasim, M. R. A Mutual Information-Driven Hybrid 1D-CNN and XGBoost Framework for Network Intrusion Detection in IoT. International Journal of Human Computing Studies 2026, 8(2), 79-93

Received: 13th Jun 2026
Revised: 28th Jul 2026
Accepted: 15th Aug 2026
Published: 28th Sep 2026



Copyright: © 2026 by the authors. Submitted for open access publication under the terms and conditions of the Creative Commons Attribution (CC BY) license (<https://creativecommons.org/licenses/by/4.0/>)

Keywords: Intrusion detection, Internet of Things, Mutual Information, 1D-CNN, XGBoost.

1. Introduction

With the emergence of Internet of Things (IoT) technology and the way it has begun to be woven into a variety of communication and industrial systems, digital interactions have come a long way. But the diversity of the devices, their dispersion, and the inherent hardware constraints of the devices in the network have made security a persistent issue. There is limited processing power and memory in IoT devices, making them an easy point of attack for security breaches. Therefore, the use of network based intrusion detection systems (NIDS) is essential as a basic defense tool for ongoing traffic monitoring and for identifying normal traffic from intrusive traffic. However, with the signature-based methods becoming less effective in the fight against unknown traffic patterns, machine learning and deep learning have become the most popular methods for designing intelligent intrusion detection mechanisms.

Although this has made great strides in this direction, there are two major problems associated with the application of data-driven models in IoT environments. First, IoT network traffic creates high-dimensional data which has redundant, repetitive or even

meaningless features that are not diagnostic. These elements increase complexity of the computation, as well as reduce the generalizability of the model, causing overfitting. Moreover, the number of optimal features in many of existing studies is decided by the manual tuning and/or some empirical assumptions, thus resulting in the biased feature selection. Second, single-algorithm models are inherently limited in their ability to offset individual weaknesses – conventional tree-based models lack the ability to capture complex, non-linear representations within raw data, and deep neural networks can be confused in the process of determining the ideal decision boundary for tabular network traffic.

In order to overcome these drawbacks, in this paper, an integrated two-stage framework for intrusion detection in IoT is presented. The proposed approach integrates the representation extraction capability of one dimensional convolutional neural network (1D-CNN) and high discriminative power of gradient boosted decision trees. The first phase is ranking the features in the traffic using a metric called Mutual Information (MI), and the best set of features is selected automatically using a geometric elbow method, which does not require manual selection. Next, those features are fed to a 1D-CNN that encodes them into a dense vector of deep features. Lastly, the XGBoost algorithm uses these deep features to carry out final classification, which is able to successfully classify intrusive traffic from regular traffic. The validation protocol strictly splits data by flows at each step of the system to assess the system under realistic conditions avoiding data leakage and information overlapping between the training and the test sets.

Accordingly, the primary contributions of this paper are summarized as follows:

- a. Development of an automated feature selection mechanism based on information theory and geometric analysis: By combining the Mutual Information metric with the elbow method we can automatically select the traffic dimensions thereby removing all manual intervention, from the dimensionality reduction process.
- b. Design of a sequential hybrid deep-ensemble architecture: Using a 1D-CNN only to extract non-linear features and then passing the resulting representation vector to XGBoost to create the best possible decision boundaries.

The rest of this paper is set out like this: Section 2 goes through the literature and related research on IoT intrusion detection. Section 3 explains the proposed method, covering data preprocessing, feature selection, the deep feature extraction architecture and the final classification model. Section 4. Discusses the experimental results and how well the model performs. In the end Section 5 wraps up this paper. Points, to future research paths.

1.1 Research Background

Shahid Allah Bakhsh and other researchers in paper [1] suggest a learning-based Intrusion Detection System (IDS) to improve security in Internet of Things (IoT) networks. This research uses three models—Feed Forward Neural Networks (FFNN) Long Short-Term Memory (LSTM) and Random Neural Networks (RandNN)—to find patterns and time-related links in network traffic. Testing these models on the CIC-IoT2022 dataset shows that the models had accuracies of 99.93%, 99.85% and 96.42% respectively showing results than other top methods, in quickly finding security problems.

Awajan in paper [2] suggests a learning-based intrusion detection system that uses a four-layer fully connected network structure. This system is made to work without depending on communication protocols. The purpose is to make it easier to put into use. The approach was tested against network attacks like Blackhole and DDoS. It got an accuracy of 93.74 percent and a precision of 93.71 percent. This shows how good it is at making IoT networks more secure. The results show performance, in keeping things safe.

Jablaoui and colleagues in paper [3] propose a deep learning-based Intrusion Detection System or IDS that mixes Convolutional Neural Networks for spatial feature extraction with Gated Recurrent Units for capturing temporal relationships. When testing this IDS on the NF-UNSW-NB15 and NF-CSE-CIC-IDS2018 datasets the results show that the hybrid IDS reaches 98.60 % accuracy on the first dataset and 97.95 % on the second which is better than using only a CNN or only a GRU. This suggests that the hybrid IDS could be a solution, for protecting modern IoT systems from cyber threats.

Haozhe Zhang in paper [4] looks at network-level security, in the Internet of Things. The research evaluates 21 network models and an ensemble learning model using the BoT-IoT dataset. The ensemble model that was created has an accuracy of 99.985% and a precision of 99.28%. This shows an improvement compared to earlier studies and proves that deep learning works well for making IoT network security better.

Sharma et al. in paper [5] propose an anomaly-based Intrusion Detection System using a Deep Neural Network with feature selection. To fix class imbalance, in the dataset the authors used Generative Adversarial Networks to create synthetic data, which raised accuracy from 84 % on the original data to 91 % on the balanced data.

Hossain in paper [6] introduces a learning-based framework for detecting threats in IoT networks. The framework uses models including 1D convolutional neural networks, long short-term memory networks, recurrent neural networks and multi-layer perceptrons. When tested on the CIC IoT-DIAD 2024 dataset the 1D convolutional neural network performed the best. It showed ability in extracting features and classifying attacks. It achieved an accuracy of 99.12% in -class classification and 99.53%, in binary anomaly detection. The 1D convolutional neural network outperformed the models in both tasks.

Jadhav in paper [7] proposes a deep learning model that combines Convolutional Neural Networks, Long Short-Term Memory and Artificial Neural Networks to enhance intrusion detection in IoT networks. Jadhav uses the BoT-IoT dataset to merge the feature extraction of CNNs, the sequential learning of LSTMs and the classification power of ANNs. This hybrid deep learning model achieves accuracy and strong reliability when detecting malicious activities, in IoT environments. I find this approach promising for intrusion detection systems.

Khorshed Alam et al. in paper [8] built a Deep Convolutional Neural Network for Network Intrusion Detection Systems. This Deep Convolutional Neural Network looks for hidden threats while keeping alarms to a minimum. On the NF-ToN-IoT dataset this Deep Convolutional Neural Network scored 99.23 % accuracy, 99.25 % precision and an F1-score of 0.9952. The strength of this Deep Convolutional Neural Network was also shown on datasets, such, as NF-UNSW-NB15 and NF-BoTIoT.

Rachana and colleagues in paper [9] created a system that mixes an LSTM and a GRU model fine-tuned with many hyper-parameters to spot intrusions, in small battery-powered IoT devices. They used a grid search to find the settings and looked closely at how traffic patterns change over time. On the CICIDS 2017 data set this model scored 95 percent accuracy 96 percent precision and a 96 percent F1-score showing that it can work well for instant threat detection.

M. Pradeep and the Research Scholar in paper [10] looking at the growth of IoT networks and the higher chances of facing cyber attacks highlight the importance of machine learning in making intrusion detection systems better. They suggest a method that uses a mix of deep learning techniques including Convolutional Neural Networks, Artificial Neural Networks, Long Short-Term Memory and Gated Recurrent Units. This method, when tested against advanced techniques shows a better balance between different performance measures compared to other methods. However the exact numbers for accuracy or precision are not given in the text. The authors only talk about how this method's better and more suitable, for complicated IoT networks.

Altaie and Hoomod in paper [11] introduce a deep learning-based intrusion detection system. This system uses a CNN-LSTM architecture. The goal is to deal with the increase in cyberattacks on devices. In this research the CNN and LSTM layers are placed one, after the other. They do not include layers. This setup helps to make the system faster and more efficient. Connected layers and softmax layers are used for classification. The model is tested on the UNSW-NB15 dataset. It is also used on a Raspberry Pi 3 for real-time detection. The model shows an accuracy of 98.78%. The recall is 98.09%. The precision is 97.88%. The F1-score is 97.99%.

Abdul Qaddos et al. in paper [12] introduce a way to detect intrusions in Internet of Things environments. This framework uses Convolutional Neural Networks and Gated Recurrent Units to find features and relationships. They employ a novel approach known as feature-weighted synthetic minority oversampling technique (FW-SMOTE) to address the issue of data imbalance. For solving data imbalance, they use feature-weighted synthetic minority oversampling technique (FW-SMOTE). They applied this approach to a dataset of IoTID20. Found that it can detect attacks with 99.60% accuracy. Further, they applied it to the UNSW-NB15 dataset. Got 99.16% accuracy. This shows that the method works well and is better than methods. It proves that this method is good, at protecting Internet of Things systems.

In paper [13] Obeidat and Shehab present a hybrid machine learning system, which combines various techniques, such as Support Vector Machines (SVM) and deep neural networks. This infrastructure will be able to manage threats like DDoS attack and e-fraud in environment. It has been tested on benchmark datasets NSL-KDD, DS2OS and IoT Botnet. The results indicated that 96.38% was detected. The high rate of threats detected demonstrates the effectiveness of the system in detecting threats. It also demonstrates a capacity to minimizing false alarms. The system improves real-time threat response making it more reliable for use, in IoT settings. SVM and deep neural networks can be used to achieve a balance between accuracy and speed. This balance is crucial in the context of the context of moving cyber threats. The findings are consistent with the ability to effectively operate in complex and dynamic environments using a hybrid approach.

In paper [14] Abdulkareem presented a deep learning (DL) framework based on CNN, BiGRU and BiLSTM to extract both spatial and temporal features from the network traffic in IoT environments. To optimize feature selection, several metaheuristic algorithms are evaluated; the model was found accurate at 98.11% by Particle Swarm Optimization (PSO) and the validation on the comprehensive dataset named as ToN-IoT was found to be a capability of the model in detecting various types of normal and adversarial traffic patterns.

Qawasmeh et al. To achieve both of these goals, the researchers in paper [15] presented a CNN-LSTM based intrusion detection system to detect a larger number of anomalies with low computation costs in IoT networks. In this system, the convolutional layer, the number of neurons in the sliding window and the weights between the neurons are adjusted according to the data and the real time input. The model could detect UNSW-NB15 with 99.1% accuracy. This is 4% better, than the baseline RNN model. The results show that the model works well in protecting the integrity and privacy of IoT systems.

2. Materials and Methods

To address all these, this research proposes a two-stage, sequential framework, which encompasses powerful preprocessing, feature selection based on information theory, learning deep non-linear representations, and classification with a boosted trees network. Figure 1 shows the overall data flow and interaction between the different system components. The information about each stage is provided below.

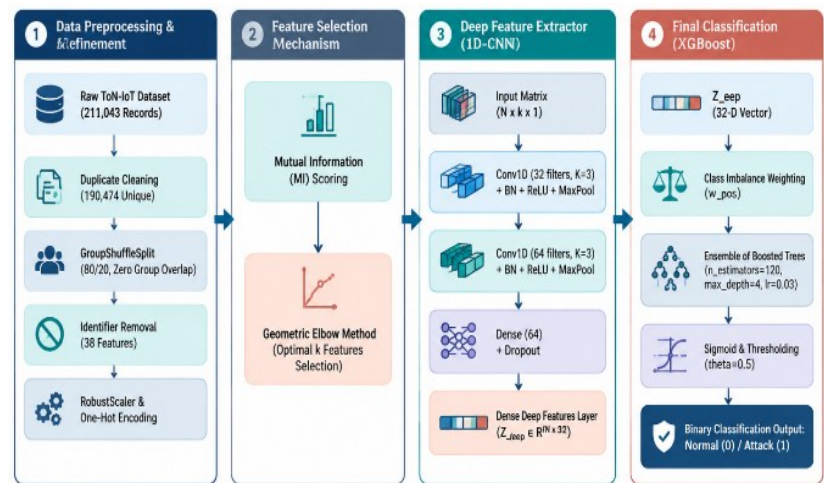


Figure 1. The proposed mutual information-driven hybrid 1D-CNN and XGBoost methodology for IoT network intrusion detection

2.1. Dataset and Data Preprocessing

2.1.1. Introduction to the ToN-IoT Dataset

In this research the known and widely accepted ToN-IoT dataset is used. This dataset was gathered in an Internet of Things environment at the Cyber Range cybersecurity laboratory of the University of New South Wales, in Canberra. The setup of the data collection system has three Devices, "Fog Layer," and "Cloud Layer" —to fully mimic real network traffic. In this study the issue is seen as a classification problem. Label 0 stands for traffic. Label 1 stands for intrusion traffic.

2.1.2. Data Preprocessing and Refinement Steps

The initial raw data consisted of 211,043 traffic records. To establish a rigorous evaluation protocol and prevent any data leakage errors, the preparation steps were implemented within an isolated pipeline as follows:

- Duplicate Record Cleaning:** In the step the goal was to stop model bias and incorrect evaluation. So identical records were found. By taking out 20,569 samples the data size went down to 190,474 unique records. The final class distribution, in this set has 42,040 samples which is 22.07% and 148,434 intrusion samples which is 77.93%.
- Data Splitting Based on Communication Flows:** To prevent data leakage that comes from sessions, between sets we divided the samples according to the communication 5-tuple, which includes IP addresses, port numbers and protocol type. We used the GroupShuffleSplit method with an 80/20 ratio. In this way 152,495 samples were put into the training set and 37,979 samples were put into the testing set. This guarantees that the group overlap coefficient is zero.
- Feature Space Reduction and Identifier Removal:** After the group split we removed identifier variables like IP addresses, ports and absolute timestamps, from the data matrix. This was done to improve generalization and stop the model from learning the identity of the devices. The input space was then reduced to 38 variables.
- Leakage-Free Feature Transformation Pipeline:** In this stage all statistical transformations were done only using the training data and then used on the test data. Missing values were filled in using the median from the training set. Twenty-four categorical variables were turned into vectors using One-Hot Encoding. Then numerical features were adjusted using the RobustScaler method and Equation (1).

$$x_{scaled} = \frac{x - Q_2(x)}{Q_3(x) - Q_1(x)} \quad (1)$$

Where Q_1 , Q_2 (median), and Q_3 are the statistical quartiles of the input feature x within the training set. Through this mechanism, the evaluation protocol remained entirely immune to any data leakage.

2.2. Mutual Information-based Feature Selection Mechanism

Given the nature of IoT network traffic and the presence of numerous non-linear features, traditional correlation methods (such as Pearson's correlation coefficient) are ineffective due to their inability to capture non-linear relationships. The approach used in this research is Mutual Information (MI) which is used to determine the level of dependence of the input feature (X) and the target variable (Y). Mutual Information is a non-parametric measure that is the amount of mutual information provided by two random variables without making any assumptions about their statistical distribution based on their probability density function defined in (2):

$$I(X; Y) = \sum_{y \in Y} \sum_{x \in X} p(x, y) \log \left(\frac{p(x, y)}{p(x)p(y)} \right) \quad (2)$$

The joint probability density function is denoted as $p(x, y)$ while the marginal functions are $p(x)$ and $p(y)$. A high $I(X; Y)$ value means that the feature has a good predictive power to distinguish intrusion classes from normal traffic. As a consequence of the data set studied being composed of various features (both numerical and categorical), a discrete mask was used during the calculation of the MI, in order to have a precise identification of variable types within the studied data set, and a good estimation of the scores, whatever the type of data.

After computing the MI scores for all features and sorting them in descending order, the Geometric Elbow Method was applied to determine the optimal number of features (k), balancing performance retention with the mitigation of the "Curse of Dimensionality." Unlike empirical thresholding, this method is entirely data-driven. Let S be the set of sorted scores; the resulting curve represents points (x_i, y_i) , where x_i is the feature index and y_i is its corresponding MI score. The optimal point is defined as the maximum vertical distance (d_i) from the line connecting the first point (x_1, y_1) to the last point (x_n, y_n) . This distance for any given point i is calculated according to Equation (3):

$$d_i = \frac{|(y_n - y_1)x_i - (x_n - x_1)y_i + x_n y_1 - y_n x_1|}{\sqrt{(y_n - y_1)^2 + (x_n - x_1)^2}} \quad (3)$$

The point yielding the maximum value of d_i is identified as the elbow point, and the corresponding number of features is selected as the optimal subset. This approach guarantees that features with the highest informational value are retained without human bias, thereby reducing the computational overhead of subsequent modeling stages.

2.3. 1D-CNN Feature Extractor

To transform the initially selected feature space into a low-dimensional, high-discriminative mapped space, a one-dimensional convolutional neural network (1D-CNN) architecture was designed as a deep feature extractor. The selected feature vector derived from the MI phase, after standard scaling via Equation (4), is transformed into an input matrix $X \in \mathbb{R}^{N \times k \times 1}$, where N is the number of samples and k is the number of primary features selected based on the geometric elbow method:

$$z_j^{(0)} = \frac{x_j - \mu_j}{\sigma_j} \quad (4)$$

where μ_j and σ_j represent the mean and standard deviation of the j -th feature in the training data, respectively. The proposed network comprises two consecutive convolutional blocks and a dense compressor mapping layer. In the l -th block ($l \in \{1, 2\}$) consisting of convolutional and pooling layers, the feature map $h^{(l)}$ is extracted by

applying 1D filters followed by the non-linear ReLU activation function, as defined in Equation (5):

$$h_i^{(l)} = \text{ReLU} \left(\sum_{m=1}^M w_m^{(l)} \cdot z_{i+m-1}^{(l-1)} + b^{(l)} \right) \quad (5)$$

where $w^{(l)}$ is the vector of convolutional kernel weights with length $K = 3$, $b^{(l)}$ represents the bias terms, and M is the number of filters in the layer, set to $M_1 = 32$ for the first layer and $M_2 = 64$ for the second. To mitigate overfitting, an L_2 regularization term with a coefficient of $\lambda_{CNN} = 0.01$ is added to the loss function according to Equation (6):

$$L_{reg} = \lambda_{CNN} \sum_l \|W^{(l)}\|_2^2 \quad (6)$$

Following each convolutional layer, Batch Normalization is applied based on Equation (7) to prevent "Internal Covariate Shift":

$$\hat{h}_i^{(l)} = \gamma \left(\frac{h_i^{(l)} - \mu_B}{\sqrt{\sigma_B^2 + \epsilon}} \right) + \beta \quad (7)$$

Subsequently, a MaxPooling1D layer with a window size of $P = 2$ is applied according to Equation (8) to reduce spatial dimensions while preserving the most salient components:

$$p_j^{(l)} = \max_{0 \leq m < P} (\hat{h}_{j \cdot P + m}^{(l)}) \quad (8)$$

Proceeding with the process, in the deep feature extraction layer (*deep_features*), the output of the second pooling layer, after Flattening, is connected to a dense layer with 64 neurons and subsequently to the key feature mapping layer, labeled *deep_features*, with $d = 32$ neurons. The output vector of this layer ($Z_{deep} \in \mathbb{R}^{N \times 32}$) is defined by Equation (9):

$$Z_{deep} = \text{ReLU}(W_d \cdot \text{Flatten}(p^{(2)}) + b_d) \quad (9)$$

threshold, λ is the L_2 regularization coefficient (with a numerical value of `reg_lambda = 5.0`), and α is the L_1 regularization coefficient (with a numerical value of `reg_alpha = 0.5`). Using the second-order Taylor expansion, the objective function at step t is simplified as shown in Equation (10):

$$\tilde{L}^{(t)} \approx \sum_{i=1}^N \left[g_i f_t(Z_{deep,i}) + \frac{1}{2} h_i f_t^2(Z_{deep,i}) \right] + \Omega(f_t) \quad (10)$$

where $g_i = \frac{\partial l(y_i, \hat{y}^{(t-1)})}{\partial \hat{y}^{(t-1)}}$ (first-order derivative or gradient) and $h_i = \frac{\partial^2 l(y_i, \hat{y}^{(t-1)})}{\partial (\hat{y}^{(t-1)})^2}$ (second-order derivative or Hessian).

Considering the imbalanced distribution of the network traffic data, to neutralize model bias toward the majority class, the weighting factor w_{pos} (`scale_pos_weight`) was calculated according to Equation (11) and applied to the gradients of positive samples (attacks):

$$w_{pos} = \frac{\sum_{i=1}^N I(y_i = 0)}{\sum_{i=1}^N I(y_i = 1)} \quad (11)$$

This weight assignment scales the loss function derivatives for minority class instances by a factor of w_{pos} :

$$g'_i = \begin{cases} g_i & \text{if } y_i = 0 \\ w_{pos} \cdot g_i & \text{if } y_i = 1 \end{cases} \quad (12)$$

The algorithm is configured on the deep feature space Z_{deep} with precisely optimized hyperparameters, such that 120 decision trees ($K_t = 120$ or `n_estimators = 120`) with a maximum depth of 4 ($D_{max} = 4$ or `max_depth = 4`) and a learning rate of 0.03 ($\eta_g = 0.03$ or `learning_rate = 0.03`) form the primary structure of gradient boosting. Furthermore, to manage tree growth and the splitting process, the minimum loss reduction parameter for a split is set to 0.5 ($\gamma = 0.5$) and the minimum child node weight is set to 6 ($w_{min} = 6$ or `min_child_weight = 6`). To ensure enhanced generalization and prevent tree correlation,

the data and feature subsampling ratios are set to 0.70 (subsample = 0.70 and colsample_bytree = 0.70), and computational acceleration during training is achieved by employing the histogram-based method (tree_method = "hist").

Finally, the final predicted probability $p_i = P(y_i = 1 | Z_{deep,i})$ is computed by applying the sigmoid function to the aggregated output of all trees, and the final predicted label $\hat{y}_i$ is determined by applying a decision threshold of $\theta = 0.5$ according to Equation (13):

$$\hat{y}_i = \begin{cases} 1 & \text{if } p_i \geq 0.5 \\ 0 & \text{if } p_i < 0.5 \end{cases} \quad (13)$$

3. Results

3.1 Experimental Results and Discussion

3.1.1 Experimental Setup

In this study, an integrated computational pipeline was used to implement all stages of data preprocessing, feature selection, design and training of the deep feature extraction architecture (1D-CNN), and the classification model (XGBoost) in Python (version 3.10). The experiments were performed using the Google Colab cloud computing platform, using an NVIDIA GPU hardware accelerator which supports CUDA. The experiments were run on Google Colab cloud computing platform, utilizing an NVIDIA GPU hardware accelerator that supports CUDA to speed up the computational operations for the convolutional layers and gradient boosting training process.

In the software aspect, data preprocessing, data management and statistical evaluation were conducted by the NumPy, Pandas and Scikit-learn libraries. The 1D-CNN feature extraction architecture was built using TensorFlow/Keras and the optimized XGBoost library was used for the last classification step. In addition, the random seed was set to a constant value (random_state = 42) in all the modules to enable complete reproducibility in all data flow splits and experimental stages.

3.1.2. Performance Evaluation Metrics

To assess the intrusion detection capability of the proposed model, the system performance is assessed using four elements of the confusion matrix: True Positive (TP), True Negative (TN), False Positive (FP) and False Negative (FN). Besides accuracy, other metrics as precision, recall and F1-score are also computed in an IoT set-up because the number of traffic samples is not balanced. Their mathematical formulations are as follows:

$$Accuracy = \frac{TP + TN}{TP + TN + FP + FN} \quad (14)$$

$$Precision = \frac{TP}{TP + FP} \quad (15)$$

$$Recall = \frac{TP}{TP + FN} \quad (16)$$

$$F1 = \frac{2 \times Precision \times Recall}{Precision + Recall} \quad (17)$$

The accuracy is the percentage of correctly identified instances over the total network traffic. The model's ability to lower false-alarms by limiting false positives is represented by precision, while recall represents the ability of the model to detect and cover attack events and not miss any true events. The last is the F1 score, which is the harmonic mean of precision and recall, which is a good and comprehensive performance measure when the classes are imbalanced.

3.1.3. Experimental Results and Performance Analysis

The empirical assessment outcomes of the proposed hybrid framework for the ToN-IoT traffic database reveal high effectiveness, convergence stability and incredible

generalization capability in the identification of cyberattacks in normal network traffic. The feature screening process was completed first to reduce the number of variables and eliminate redundant ones in the evaluation process. In this step, the mutual information mechanism was used in conjunction with the geometric elbow criterion to facilitate automated identification of top 18 basic features, and the importance ranks of these features were shown in Figure 2. According to this evaluation, the volumetric statistical feature `src_ip_bytes` with a mutual information score of 0.4042 and the communication protocol `proto` with a score of 0.3869 exhibited the highest degree of non-linear dependency with the traffic labels. Furthermore, Domain Name System (DNS)-related variables, such as `dns_query` (0.3274), `dns_rejected` (0.3012), and response flags including `dns_RA` and `dns_AA`, were ranked among the top 10 features, highlighting the pivotal role of anomalous DNS application-layer behaviors in detecting IoT network intrusion events. The elimination of low-impact features not only refined the input vector but also established an optimal representation space for the precise training of the neural network.

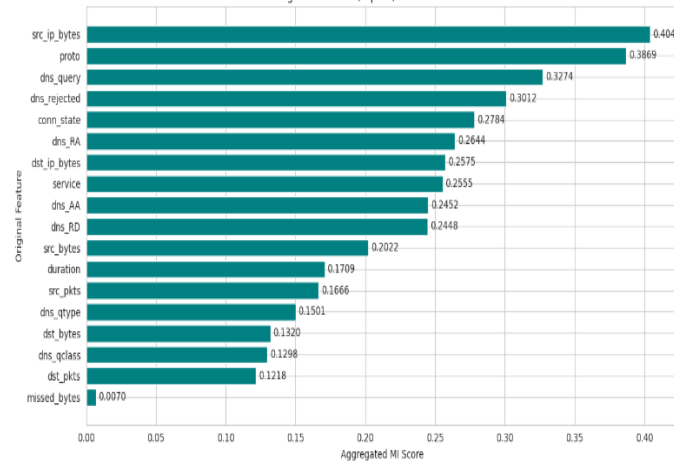


Figure 2. Ranking of the top 18 selected features based on cumulative Mutual Information (MI) scores.

Following the selection of the top 18 features and the refinement of the input space, these features were fed into the one-dimensional convolutional neural network (1D-CNN) architecture in the subsequent stage to extract deep spatial patterns and non-linear structural dependencies inherent in network traffic. The training process of this feature extractor was initiated with a batch size of 128, a maximum of 30 epochs, a binary cross-entropy loss function, and the Adam optimizer initialized with a learning rate of 10^{-3} . To prevent overfitting and guarantee convergence stability, an Early Stopping mechanism monitoring the validation loss (with a patience of 5 epochs and restoring the best model weights) was utilized, along with an adaptive learning rate reduction strategy (with a reduction factor of 0.5, a patience of 2 epochs, and a minimum threshold of 10^{-5}) deployed on 10% of the training data. Consequently, the model achieved stable convergence at the 21st epoch. As depicted in the accuracy and loss trajectories in Figure 3, the validation loss decreased from an initial value of 0.1742 in the first epoch to a minimum of 0.0344 at epoch 16, while the validation accuracy progressively improved from 98.65% to over 99.70%, closely mirroring the training trajectory. The close alignment and absence of divergence between these two curves demonstrate the efficacy of the batch normalization layers, Dropout, and L_2 regularization penalty in comprehensively mitigating overfitting. Upon the stabilization of network training, the compact 32-dimensional representation vector (Z_{deep}) was extracted from the deep feature layer to serve as a high-quality latent representation for the final classification phase.

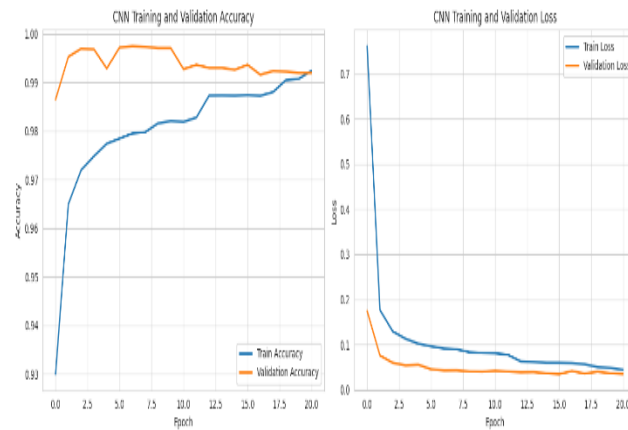


Figure 3. Training and validation dynamics of the 1D-CNN architecture: (a) accuracy trajectories, (b) declining trend of the loss function.

Following the extraction of these deep representation vectors, the XGBoost decision tree boosting model was subsequently deployed on these representations in the next stage to establish an optimal, robust, and discriminative decision boundary between normal and intrusive traffic. The convergence behavior of the XGBoost classifier on these deep vectors across 120 boosting iterations is illustrated via the log-loss trajectory in Figure 4. As demonstrated by this curve, the log-loss on the test data exhibits a strictly monotonic and decreasing trend, declining synchronously with the training trajectory from values exceeding 0.66 to below 0.04 by the end of the 120th iteration without any noticeable generalization gap. This behavior highlights the superior discriminative capability of the deep feature vectors generated by the 1D-CNN, as well as the optimal boundary-construction efficacy of the gradient boosting model.

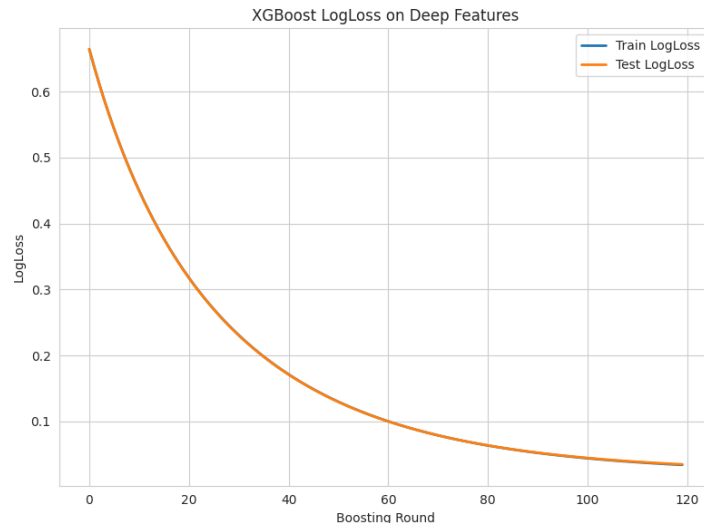


Figure 4. Log-loss reduction trend of the XGBoost model across 120 boosting iterations.

Then, after the decision boundaries were consolidated using the boosting algorithm, a quantitative assessment of the framework was performed on 37,979 independent flow instances of the test set for a thorough validation of the operational reliability and to enable the evaluation of the system's behavior in a realistic deployment. The final decision making process generated the following confusion matrix (Figure 5). Of these 195 false positive events, 7,883 traffic instances were classified correctly of 8,078 normal traffic instances, as shown in Figure 5. In the case of intrusive traffic, 29,901 attack occurrences were identified; only 219 were false negatives. The false negative rate is extremely small

due to using the adaptive gradient weighting in boosting which is an important operational benefit for IoT network security.

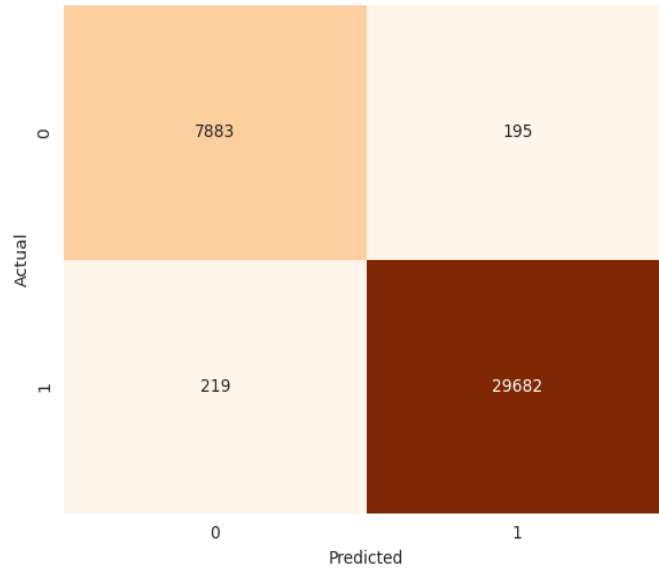


Figure 5. Confusion matrix of the hybrid 1D-CNN + XGBoost model on the test dataset.

To evaluate how the system behaves with classification thresholds the Receiver Operating Characteristic (ROC) curve is shown in Figure 6. The fast movement of this curve toward the left corner shows that the system has a very high true positive rate and a very low false positive rate. This is a feature for real-time monitoring of IoT network flows because it helps reduce false alarms a lot. Getting an Area Under the Curve (AUC) of 0.9986 shows that the model has ability to tell the difference, between classes and makes stable decisions even when the probability threshold changes. This proves that the hybrid approach works well.

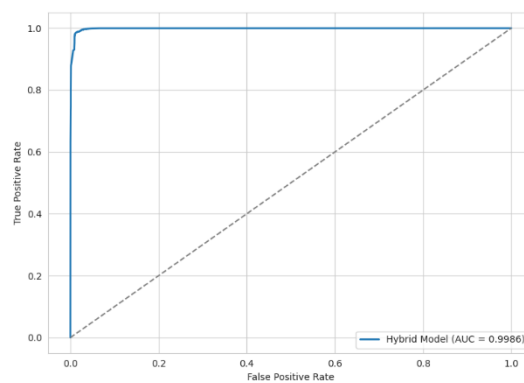


Figure 6. Receiver Operating Characteristic curve and Area Under the Curve of the proposed hybrid model.

A detailed statistical analysis is performed about the accuracy, precision, recall and F1 score for each individual class and the result is summarized in Table 1. The overall accuracy of the proposed framework was 0.9891 when evaluated on 37,979 independent flow instances of the test dataset. For the intrusive traffic class, the model achieved a precision of 0.9935, a recall of 0.9927 and an F1-score of 0.9931 in class-specific discrimination. The strong relationship between precision and recall in this class shows that there were no unnecessarily high false alarm rates when improving the intrusion detection capability.

A precision of 0.9730 and a recall of 0.9759 were obtained by the model on the normal traffic class, showing the balanced performance of the algorithm, where it recognizes benign network patterns. Although there was an imbalance of 29,901 attack samples and

8,078 normal samples, the macro-averaged (Macro Avg) scores for precision, recall, and the F1-score were 0.9832, 0.9843, and 0.9837 respectively. A close convergence of the two averages in this case, macro average 0.9891 and the weighted average, suggests that there is no majority class bias in the proposed framework, and that it maintains its discriminative power in a well-balanced manner across both classes.

Table 1. Detailed classification performance report of the proposed framework on the test dataset

Traffic Class	Precision	Recall	F1-Score	Support
Normal Traffic (Class 0)	0.9730	0.9759	0.9744	8,078
Intrusion Traffic (Class 1)	0.9935	0.9927	0.9931	29,901
Overall Accuracy	—	—	0.9891	37,979
Macro Avg	0.9832	0.9843	0.9837	37,979
Weighted Avg	0.9891	0.9891	0.9891	37,979

3.2 Comparative Evaluation

To assess the efficacy of the proposed framework for network intrusion detection in IoT environments, its performance on the ToN-IoT dataset was compared against previous baseline studies. The comparative evaluation results are summarized in Table 2.

Table 2. Accuracy comparison of the proposed framework with related studies on the ToN-IoT dataset

Study	Methodology / Employed Architecture	Dataset	Accuracy
Study [16]	Two-stage automated system based on enhanced LSTM	ToN-IoT	96.35%
Study [17]	Integrated deep learning based on Autoencoder and DNN	ToN-IoT	98.00%
Study [18]	Hybrid Decision Tree and Convolutional Neural Network (CNN)	ToN-IoT	96.80%
Proposed Framework	Hybrid Mutual Information, CNN, and XGBoost	ToN-IoT	98.91%

The outcome of the proposed framework shown in the Table 2 proves that the performance of the proposed framework is better than previous studies with an overall accuracy of 98.91%. Studies [16] and [18] show that the use of recurrent layers on heterogeneous flow data is a problem, and the generalization problem of traditional decision trees limits their classification accuracy to 96.35% and 96.80%, respectively. The proposed method overcomes these drawbacks by making efficient use of automated deep feature extraction and an advanced gradient boosting algorithm. Moreover, the improvement in accuracy of approximately 1% over the accuracy of Study [17] (98.00%)

is due to the optimum features space obtained using the mutual information driven filtering of irrelevant and noisy features which enables better delineation of the decision boundary. In conclusion, the combination of mutual information-based feature selection, extraction of abstract representations from the input, and robust ensemble classification, has demonstrated the increased stability and effectiveness of the proposed framework for IDS in IoT networks.

4. Discussion

The experimental findings demonstrate that the proposed Mutual Information-driven hybrid 1D-CNN and XGBoost framework provides a highly effective approach for IoT network intrusion detection by integrating dimensionality reduction, non-linear feature extraction, and robust ensemble classification within a single sequential pipeline. The selection of the top 18 features using Mutual Information and the geometric elbow criterion indicates that reducing redundant and low-impact variables can preserve the most discriminative information while lowering the complexity of subsequent learning stages. The prominence of features such as `src_ip_bytes`, `proto`, and several DNS-related variables further suggests that volumetric traffic characteristics, communication protocols, and anomalous application-layer behaviors play an important role in distinguishing malicious from benign IoT traffic. The 1D-CNN training process showed stable convergence, with validation accuracy exceeding 99.70% and validation loss declining to 0.0344, while the close alignment between training and validation curves indicates that the use of batch normalization, dropout, and regularization effectively limited overfitting. In the classification stage, XGBoost maintained a steadily decreasing log-loss without a noticeable generalization gap, demonstrating that the deep feature representations extracted by the 1D-CNN provided a highly discriminative latent space for constructing decision boundaries. The final evaluation on 37,979 independent test flows produced an overall accuracy of 98.91%, an AUC of 0.9986, and intrusion-class precision, recall, and F1-score values of 99.35%, 99.27%, and 99.31%, respectively, indicating a strong balance between attack detection and false-alarm control. Moreover, the macro-averaged and weighted performance scores were closely aligned, suggesting that the model was not dominated by the majority class despite the imbalance between intrusion and normal samples. Compared with previous ToN-IoT-based studies, the proposed framework achieved higher accuracy than the enhanced LSTM, Autoencoder-DNN, and hybrid Decision Tree-CNN baselines, which supports the effectiveness of combining information-theoretic feature selection with deep representation learning and gradient boosting. These results indicate that the main advantage of the proposed framework lies not only in high predictive performance, but also in its ability to reduce irrelevant input dimensions, capture complex non-linear traffic patterns, and maintain balanced discrimination across heterogeneous IoT traffic classes.

5. Conclusion

Internet of Things (IoT) environments are widespread in critical infrastructure, greatly increasing the attack surface, especially when dealing with complex and varied network traffic, which makes traditional intrusion detection systems (IDS) less effective. This paper presented an intelligent hybrid model based on one-dimensional convolutional neural network and extreme gradient boosting that combines both models to provide robust intrusion detection in IoT networks using mutual information. The architecture alleviated computational overhead, high dimensionality by mutual information based feature selection, captured non-linear structure representation by optimized convolutional layers and delineated resilient decision boundary using an engineered classifier that was able to effectively deal with class imbalance.

The proposed methodology was able to outperform the other methodologies through rigorous empirical evaluations on the benchmark ToN-IoT dataset for 37,979 independent test flow instances. The overall accuracy was 98.91%, precision was 99.35%, recall was 99.27%, F1F₁ score in the intrusion class was 99.31% and the area under receiver operating characteristic curve was 0.9986. The framework suggested in this paper is able

to maintain the detection rate balance for both normal and intrusive traffic classes, while not having majority class bias, as is shown in the comparative analysis with baseline studies.

On the basis of this research direction one can imagine several directions to take future research. Expanding to multi-class classification to distinguish particular attack vectors, like DDoS, ransomware and new variants of malware will increase its operational use. Additionally, it is important to compress and lighten the deep feature extraction layers for real-time use on the edge nodes and embedded devices with limited resources, like the IoT gateway devices. Moreover, compressing and lightening the deep feature extraction layers for real time implementation on limited resources edge nodes and IoT gateway is a critical step towards the practical implementation. Last but not least, such an architecture can be complemented by federated and online learning approaches, enabling the adaptation to zero-day threats while maintaining data privacy across distributed environments.

REFERENCES

- [1] S. A. Bakhsh, M. A. Khan, F. Ahmed, M. S. Alshehri, H. Ali, and J. Ahmad, "Enhancing IoT network security through deep learning-powered intrusion detection system," *Internet of Things*, vol. 24, p. 100936, 2023.
- [2] A. Awajan, "A novel deep learning-based intrusion detection system for IoT networks," *Computers*, vol. 12, no. 2, p. 34, 2023.
- [3] R. Jablaoui, O. Cheikhrouhou, M. Hamdi, and N. Liouane, "Deep learning enabled intrusion detection system for IoT security: R. Jablaoui et al.," *EURASIP Journal on Wireless Communications and Networking*, vol. 2025, no. 1, p. 66, 2025.
- [4] H. Zhang, "Development of an intelligent intrusion detection system for IoT networks using deep learning," *Discover Internet of Things*, vol. 5, no. 1, p. 74, 2025.
- [5] B. Sharma, L. Sharma, C. Lal, and S. Roy, "Anomaly based network intrusion detection for IoT attacks using deep learning technique," *Computers and Electrical Engineering*, vol. 107, p. 108626, 2023.
- [6] M. A. Hossain, "Deep learning-based intrusion detection for IoT networks: A scalable and efficient approach," *EURASIP Journal on Information Security*, vol. 2025, no. 1, p. 28, 2025.
- [7] P. M. Jadhav, "Intrusion detection using IoT with deep learning," *International Journal of Scientific and Advanced Technology*, vol. 16, no. 3, 2025.
- [8] K. Alam, M. F. Monir, Z. Hassan, and M. T. Habib, "Optimizing IoT network intrusion detection: A deep learning approach," in *Proc. 2024 7th Conf. Cloud and Internet of Things (CIoT)*, Oct. 2024, pp. 1–5.
- [9] P. Rachana, M. Talakoti, and V. S. N. Reddy, "Development of hybrid intrusion detection systems for IoT enabled devices utilizing resource constraint learning frameworks," *Journal of Smart Internet of Things*, vol. 2024, no. 1, pp. 60–76, 2024.
- [10] M. Pradeep and S. Gopalakrishnan, "Enhancing intrusion detection systems in IoT networks: A hybrid approach using CNN, ANN, LSTM, GRU for improved security," in *Proc. 2024 8th Int. Conf. Inventive Systems and Control (ICISC)*, Jul. 2024, pp. 487–492.
- [11] R. H. Altaie and H. K. Hoomod, "An intrusion detection system using a hybrid lightweight deep learning algorithm," *Engineering, Technology & Applied Science Research*, vol. 14, no. 5, pp. 16740–16743, 2024.
- [12] A. Qaddos, M. U. Yaseen, A. S. Al-Shamayleh, M. Imran, A. Akhunzada, and S. Z. Alharthi, "A novel intrusion detection framework for optimizing IoT security," *Scientific Reports*, vol. 14, no. 1, p. 21789, 2024.
- [13] M. Obeidat and R. Shehab, "Network-based intrusion detection in IoT environments using hybrid machine learning techniques," *Babylonian Journal of Networking*, vol. 2025, pp. 116–125, 2025.
- [14] A. B. Abdulkareem, "Advances in IoT intrusion detection: Deploying hybrid deep learning and metaheuristic algorithms for optimal feature selection," *Ingénierie des Systèmes d'Information*, vol. 30, no. 4, p. 1027, 2025.
- [15] S. Qawasmeh, A. Habboush, B. Elzaghmouri, Q. Kharma, and D. A. Albalawneh, "Hybrid convolutional neural network-based intrusion detection system for secure IoT networks," *Tikrit Journal of Engineering Sciences*, vol. 32, no. SP1, pp. 1–11, 2025.
- [16] R. A. Elsayed, R. A. Hamada, M. I. Abdalla, and S. A. Elsaid, "Securing IoT and SDN systems using deep-learning based automatic intrusion detection," *Ain Shams Engineering Journal*, vol. 14, no. 10, p. 102211, 2023.
- [17] H. Kamal and M. Mashaly, "Hybrid deep learning-based autoencoder-DNN model for intelligent intrusion detection system in IoT networks," in *Proc. 2025 15th Int. Conf. Electrical Engineering (ICEENG)*, May 2025, pp. 1–6.

-
- [18] S. M. Ahmed, S. S. Islam, and M. S. Ullah, "Hybrid machine learning and deep learning approaches for anomaly detection using KD99 and TON_IoT datasets," in *Proc. 2025 Int. Conf. Electrical, Computer and Communication Engineering (ECCE)*, Feb. 2025, pp. 1–6.