I.J. Wireless and Microwave Technologies, 2026, 5, 66-88
Published Online on October 8, 2026 by MECS Press (http://www.mecs-press.org/)
DOI: 10.5815/ijwmt.2026.05.05

A Lightweight CNN-GRU Based Intrusion Detection Model with Mutual Information Feature Selection for IoT Edge Devices

Safwan Ishrak
Department of Computer Science and Telecommunication Engineering, Noakhali Science and Technology University, Noakhali, Bangladesh
E-mail: safwanishraksiam@gmail.com
ORCID iD: https://orcid.org/0009-0005-5511-3381

Puja Dhar
Department of Computer Science and Telecommunication Engineering, Noakhali Science and Technology University, Noakhali, Bangladesh
E-mail: pujadhar.here@gmail.com
ORCID iD: https://orcid.org/ 0009-0008-0740-280X

Md. Abdul Wahab
Department of Computer Science and Engineering, BGC Trust University Bangladesh, Chattogram, Bangladesh
E-mail: wahab@bgctub.ac.bd
ORCID iD: https://orcid.org/0000-0001-7256-7725

ARM Mahamudul Hasan Rana*
Department of Computer Science and Telecommunication Engineering, Noakhali Science and Technology University, Noakhali, Bangladesh
E-mail: m.hasan@nstu.edu.bd
ORCID iD: https://orcid.org/0000-0002-2539-5307
*Corresponding Author

Ratnadip Kuri
Department of Computer Science and Telecommunication Engineering, Noakhali Science and Technology University, Noakhali, Bangladesh
E-mail: ratnadip@nstu.edu.bd
ORCID iD: https://orcid.org/0000-0001-7335-7922

Humayun Kabir
Department of Computer Science and Telecommunication Engineering, Noakhali Science and Technology University, Noakhali, Bangladesh
E-mail: kabir@nstu.edu.bd
ORCID iD: https://orcid.org/0009-0000-3765-0662

Received: June 16, 2026; Revised: July 16, 2026; Accepted: July 24, 2026; Published: October 8, 2026

Abstract: The rapid expansion of memory and resource-constrained IoT devices has enormously increased vulnerability to cyber intrusions. Although deep learning-based intrusion detection systems (IDS) aim to improve intrusion precision, this improvement comes at the cost of increased inference latency and resource utilization. In this paper, we propose a lightweight CNN-GRU-based IDS model that utilizes mutual information-based feature selection to reduce input dimensionality, retaining the most informative features. The model is validated using four popular IoT security datasets: BoT-IoT, ToN-IoT, Edge-IIoTSeT and NSL-KDD. We employ SMOTE to reduce class imbalance in the training dataset for classifying both major and minor attacks. We achieve accuracies of 99.31%, 98.80%, 96.73%, and 94.20% on BoT-IoT, NSL-KDD, ToN-IoT and Edge-IIoTSeT respectively. Since inference latency is a critical requirement for resource-constrained IoT devices, the proposed model achieves inference times of 0.064 ms, 0.086 ms, 0.078 ms, and 0.073 ms on the respective datasets. These results demonstrate that the proposed IDS provides an effective balance between detection

This work is open access and licensed under the Creative Commons CC BY 4.0 License.

performance and computational efficiency for real-time IoT applications. In future we will focus on validating the proposed framework in real-world IoT deployment scenarios.

Index Terms: IoT Security, Intrusion Detection System, CNN-GRU, Mutual Information Feature Selection, SMOTE

1. Introduction

In a world increasingly driven by intelligent systems, the Internet of Things (IoT) is considered a big transformative force that connects billions of devices to sense, compute, and act. The possibilities for automation and improved usability through smart grid systems, autonomous transportation, industrial robotics, and precision medicine are significantly enhanced by IoT systems [1]. These devices operate across various domains, generating huge amounts of data with minimal human intervention, often depending on seamless integration with cloud platforms and edge computing.

However, the large-scale deployment of IoT devices introduces two closely related challenges. First, most IoT devices operate under strict resource constraints, including limited computational capability, memory, storage capacity, and battery lifetime [2]. Second, the highly interconnected nature of IoT environments increases the attack surface, making devices attractive targets for cyber adversaries seeking to disrupt critical services or compromise sensitive information [3,4] . Among the various threats facing IoT ecosystems, cyber intrusions remain one of the most persistent and evolving challenges.

An Intrusion Detection System (IDS) is essential to identify unauthorized access and malicious activities in IoT devices. Traditional IDS [5] models usually fail in these decentralized and resource-constrained environments. Therefore, Intrusion Detection Systems (IDSs) that monitor and analyze network traffic to detect malicious activity are becoming indispensable. To support real-time deployment in resource-constrained IoT environments, lightweight IDS models are required. Lightweight IDS are generally characterized by low computational, storage, and energy requirements while maintaining effective intrusion detection capability [2].

While many approaches have been proposed for IDSs, the actual deployment in an IoT environment is not straightforward. Deep-Learning(DL) based intrusion detection systems have offered superior detection capabilities, especially with hybrid architectures of CNN and RNN, whereby spatial and temporal correlations in network data are captured [6,7] . However, memory-intensive applications require prolonged training and inference times, and have a multitude of input features, making them inefficient for lightweight IoT edge devices [8–10] .

This gap between accuracy and efficiency presents a real-world deployment dilemma. In practice, IoT-based IDS must be able to detect intrusions in a split second or just a few milliseconds, and do so on devices that have minimal RAM to work with, too little computational power, and very little battery life to spare for other consuming tasks. Besides, what might be in datasets are extraneous or redundant features that justify the increase in the model's size but do little to improve its performance. Hence, the primary motivation for the study is to build a lightweight IDS that balances precision with practicality — one that does not just detect threats accurately, but does so quickly and with minimal computational overhead [11,12].

In this study, we aim to mitigate this gap by proposing a lightweight CNN-GRU hybrid IDS that is customized for the IoT environment. The following are the goals of this study.

- A hybrid deep learning framework that can effectively extract the spatial as well as temporal features of network traffic to drive better detection of advanced cyber-attacks.
- A feature engineering process to reduce input dimensionality and optimize the relevance of selected features to make models more efficient and faster in inference.
- A focus given to the model's robustness to non-uniform data distributions, with enhanced generalizability across a broad spectrum of attack classes, i.e., rare or minority attacks.
- Comprehensive testing of performance across a wide variety of benchmark IoT datasets, making it evident that the model possesses latency-friendly inference, scalability, and usability in real-time edge environments.

The framework is fairly put through the wringer with four popular benchmark datasets, namely BoT-IoT [13], ToN-IoT [14], Edge-IIoTSeT [15], and NSL-KDD [16], each representing a distinct IoT threat environment.

Real-world IoT data streams are unpredictable, non-stationary, and exhibit diverse statistical properties. Higher data dimension features would bloat the model and also slow down inference, making it difficult to get a quick response. In this paper, we address these challenges by introducing a unified, lightweight, and generalized CNN-GRU-based intrusion detection system, augmented with mutual information-based feature selection to retain the most relevant features, reduce redundancy, and enhance inference efficiency—providing a small step toward bridging the gap between high performance and real-world operational security deployment in IoT environments. Our proposed model is more explainable and interpretable on critical infrastructure and consumer-facing devices.

The other parts of the research work are as follows: Section 2 provides related work, while section 3 briefly explains

the core of the research work, i.e., the proposed intrusion detection methodology. Section 4 includes results and their subsequent discussions, and then come the concluding remarks in section 5 concerning the research work.

2. Literature Review

As introduced, the fast evolution of IoT networks has brought about a double-edged scenario—great opportunity and yet unheralded vulnerabilities. The pressing need is to have competent intrusion detection systems (IDS) designed to be lightweight yet provide adequate accuracy. Deep learning has indeed come of age here, especially for detecting complex, never-seen-before attack patterns, yet introducing these solutions in the constrained environment of IoT is still a big issue. Over the years, numerous hybrid deep learning architectures have been proposed to address this issue. Each presents a different kind of solution, some focusing on maximization of detection accuracy while others take speed or lightweight deployment into consideration.

Early deep learning-based approaches were presented by Yin et al. [6], who then used recurrent neural networks (RNN) to create an intrusion detection system using the NSL-KDD dataset. Their result was better than classical machine learning techniques, such as SVM or Random Forest, in both binary and multi-class classifications. However, in exchange for the performance improvement, the model suffered training inefficiencies and instability, owing to the aggravating difficulty of vanishing gradients. For a more stable and efficient model, Mounika et al. suggested a hybrid framework of modified CNN with Local Outlier Factor-based data balancing and Recursive Feature Elimination for feature selection [17]. When trained using the NSL-KDD dataset, their system generated an accuracy of 99.3% while reducing false alarms to 3.02%, thereby beating the DNN, ResNet, and GBDT SMOTE baselines. Although the model showed promising results, the evaluation was confined to NSL-KDD, and there was no attempt to evaluate the inference performance of the model on real-time or resource-constrained systems.

Following the temporal-learning path, Sinha and Manollas [18] introduced an architecture combining 1D CNN with Bi-LSTM for capturing both spatial and temporal correlations in the data. Their model exhibited topnotch accuracy on both NSL-KDD and UNSW-NB15 datasets, demonstrating that hybrid networks can indeed surpass standalone models. Yet, deployment options were not discussed, nor was the generalizability of the approach to different traffic patterns. Pushing the boundaries still, Hnamte and Hussain [19] proposed a deep CNN-BiLSTM model (DCNNBiLSTM), which attained near-perfect detection scores on the CICIDS2018 and Edge-IIoTSet datasets. While an achievement commendable indeed, the complexity of their proposed architecture questioned its appropriateness in real-time settings or resource-constrained environments.

In contrast, Catillo et al. [20] followed somewhat a different path, using deep autoencoders for their purposes. On the N-BaIoT dataset, their AE-based IDS uplifted a recall of 0.9997 with practically zero false positives. This semi-supervised approach is great for zero-day threats but its scope remains limited to a solitary dataset and it did not venture into concerns of scalability across distributed IoT infrastructures. Popoola et al. [21] wanted to strike a balance of accuracy and compactness in their LAE-BiLSTM model since it had remarkable accuracy and yet reduced data size by more than 90%. Their method would have been an answer to the memory issues in the IoT devices. However, since only the BoT-IoT dataset was used, the model's cross-platform performance remains unclear.

Another approach was taken by Firas et al. [22], who developed a hybrid IDS applying CNN and XGBoost with very special attention to IoT traffic. CNN interprets the spatial patterns from input features, while XGBoost is regarded as a very strong and efficient classifier. The model performs really well on the NF-bot-IoT dataset(98.76%), with very low false positive rates. From this point of view, the model should be tested on other datasets, which would limit the usages of the model in different IoT environments.

Cao et al. [9] came up with a CNN-GRU architecture that was further enhanced by feature selection and resampling procedures to counteract the effects of feature redundancy and data imbalance. Their model had very good performance on different datasets, namely, NSL-KDD(99.69%) and CICIDS2017(99.65%); however, with a high number of parameters, it casts prospects about its deployment in an IoT edge environment.

Likewise, Qaddos et al. [23] enhanced the CNN-GRU by using Feature-Weighted SMOTE to counteract the class imbalance for the IoTID20 dataset. Their model produced great results on both IoTID20(99.60%) and UNSW-NB15(99.16%) datasets. However, the complexity that came with oversampling also added to processing overheads, which may have a negative impact on real-time performance on constrained devices.

Jouhari et al. [24] explored yet another angle—using Chi-Square-based feature selection applied to a lightweight CNN-BiLSTM framework. This setup was well suited to resource-limited environments and performed well on UNSW-NB15(97.90%). Nevertheless, their study cited further work on cost reduction and adaptation to dynamic traffic.

Lo et al. [25] introduced E-GraphSAGE, a GNN-based IDS for IoT networks that considers an IoT network as a graph domain with nodes representing hosts and edges representing flow records. Learning both from edge features and graph structures, the approach reached F1-scores of up to 1.00 on BoT-IoT and ToNIoT datasets. Nonetheless, the graph operations may be susceptible to scalability and deployment issues as the environment becomes resource limited.

Singh and Kazmi [26] proposed FusionSec-IoT, a federated deep-learning framework based on CNN, LSTM, and FCNN models on multi-view network traffic. By combining the hybrid feature selection with a reinforcement-learning approach, they were able to increase detection accuracy(97.2%) and handle zero-day attacks. At the same time, the frame-

work protects privacy and allows for edge deployment; however, the integration of GNNs and meta-learning could have generated some overhead for the severely constrained devices.

The summary of these and other related works can be seen in Table 1 highlighting features, limitations, and performance values across different datasets.

Table 1. Summary of the Works related to Intrusion Detection Based on Hybrid Deep Learning.

Papers	Features	Research Gaps	Accuracy (%)
Yin et al. [6]	RNN-based model captures sequential patterns, outperforms traditional ML	Limited to NSL-KDD; suffers from long training and vanishing gradients	83.28 (KDDTest+), 68.55 (KDDTest-21), 99.81 (Train+)
Sinha & Manollas [18]	CNN-BiLSTM hybrid; improves detection and reduces false positives	Needs real-world evaluation; lacks efficiency tuning	99.22 (NSL-KDD), 98.88 (UNSW-NB15)
Mounika et al. [17]	Combines modified CNN with Local Outlier Factor for dataset balancing and RFE for optimal feature selection; achieves low false alarm rate and high classification accuracy.	Limited to NSL-KDD; lacks validation on modern or real-world datasets; does not explore resource constrained deployment feasibility.	99.30% (NSL-KDD)
Hnamte & Hussain [19]	DCNN-BiLSTM; very high detection on CICIDS and Edge-IIoT	Too deep for real-time use; high training cost	100.00 (CICIDS), 99.64 (Edge-IIoT)
Catillo et al. [20]	Autoencoder reduces dimensionality while keeping accuracy high	Only tested on N-BaIoT; no cross-dataset validation	99.97 (N-BaIoT)
Popoola et al. [21]	LAE-BiLSTM reduces 91% of storage; good for IoT memory limits	Limited dataset; needs more reduction method exploration	100.00 (BoT-IoT)
Firas et al. [22]	CNN-XGBoost hybrid for spatial features and fast classification	Lacks evaluation on diverse datasets	98.76 (NFbot-IoT)
Cao et al. [9]	CNN-GRU with sampling and FS; handles imbalance	Too many parameters; not deployable on edge devices	99.69 (NSL-KDD), 86.25 (UNSW), 99.65 (CICIDS)
Qaddos et al. [23]	CNN-GRU + FW-SMOTE; better minority class handling	SMOTE increases processing time; not real-time ready	99.60 (IoTID20), 99.16 (UNSW)
Jouhari et al. [24]	Chi-Square FS + CNN-BiLSTM; faster inference	Needs further cost optimization	97.90 (UNSW-NB15)
Lo et al. [25]	E-GraphSAGE: GNN using edge and host flows	Graph ops may not scale for low-end devices	F1: 100 (BoT-IoT), 87 (ToN-IoT)
Singh & Kazmi [26]	CNN+LSTM+FCNN w/ federated learning and RL	Meta-learning adds compute overhead on edge	97.2 (Custom IoT)

Table 1 illustrates the evolution of lightweight IoT intrusion detection research from 2017 to 2025. Earlier studies primarily focused on improving detection accuracy through increasingly sophisticated deep learning architectures, such as RNN, CNN, and CNN-BiLSTM based models. Subsequent research introduced hybrid CNN-GRU architectures combined with feature selection and data balancing techniques to reduce feature redundancy and improve class imbalance handling. More recent studies have explored graph neural networks, federated learning, and advanced feature selection methods to improve scalability, privacy, and detection capability. Despite these improvements, many existing approaches still rely on single-dataset evaluation, involve computationally expensive architectures, or provide limited analysis of inference latency and deployment feasibility in resource-constrained IoT environments. Motivated by these limitations, this work proposes a lightweight CNN-GRU-based IDS with mutual information-based feature selection and evaluates it across four benchmark datasets to achieve an effective balance between detection performance and practical deployment efficiency. The following section elaborates on the methodology of the proposed system.

3. Methodology

3.1. Proposed CNN-GRU Based Intrusion Detection Framework

The proposed framework introduces a light-weight and efficient deep learning IDS for resource-constrained IoT environments. The core methodology involves mutual information-based feature extraction and a hybrid CNN-GRU frame-work to enhance detection rate without spending much computing overhead. The work is evaluated on four

standard benchmark datasets—Edge-IIoTset, Bot-IoT, ToN-IoT, and NSL-KDD—and each dataset offers unique IoT-focused at-tack signatures.

The contributions of the proposed framework lie in the following aspects:

- **Hybrid Temporal-Spatial Modeling:** The spatial feature extraction is handled by one-dimensional convolutional neural networks, while the GRU is tasked to learn the temporal relationships, and therefore, the model successfully learns sequential and local attack patterns.
- **Mutual Information-based Feature Optimization:** The mutual information approach retains only the most informative features, reducing input dimensions and enhancing model efficiency without degrading accuracy. Feature selection criteria are dynamically executed for each dataset to attain a compromise between models' overall performance and resource consumption.
- **Class Imbalance Management with SMOTE:** The SMOTE (Synthetic Minority Oversampling Technique) is utilized for managing the class imbalance of real-world datasets in IoT scenarios. This avoids the model's inclination toward majority classes and improves generalizability.

Fig. 1 illustrates the flow diagram for the proposed IDS architecture, with all preprocessing, feature selection, and deep learning methods. This integrated solution to the dual needs of detection accuracy and computation scalability renders it an appealing solution to modern IoT network security.

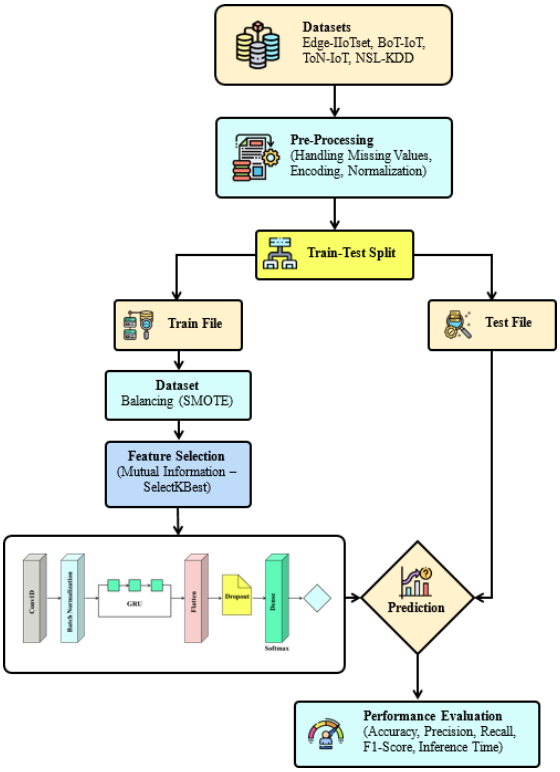


Fig. 1. Framework of proposed workflow.

3.2. Datasets

The ability of the proposed CNN-GRU-based IDS is assessed using four publicly available, popular, bench-mark datasets. These are Edge-IIoTset, Bot-IoT, ToN-IoT, and NSL-KDD. We choose these datasets because they are well-known IoT cyber-security datasets and represent both modern and traditional network threats.

Edge-IIoTset: The Edge-IIoTset dataset was created by Ferrag et al. in 2022 [15], and it is said to be a realistic cyber-security dataset for IoT and IIoT environments. The DNN-EdgeIIoT-dataset segment is employed in this re-search. There are around 1.9 million records, both normal and malicious types. The dataset contains 62 features, with 14 attack types such as DoS, DDoS, malware, injection, and reconnaissance. The diversification of this dataset makes it fit for the training and testing of a deep learning-based intrusion detection system in a complicated IoT environment. Class distribution in Edge-IIoTSet dataset is shown in Table 2.

Table 2. Class distribution in the Edge-IIoTSet dataset.

Class Label	Count
Normal	1,363,998
DDoS UDP	121,567
DDoS ICMP	67,939
SQL injection	50,826
DDoS TCP	50,062
Vulnerability scanner	50,026
Password	49,933
DDoS HTTP	48,544
Uploading	36,807
Backdoor	24,026
Port Scanning	19,977
XSS	15,066
Ransomware	9,689
Fingerprinting	853
MITM	358

Bot-IoT: The Bot-IoT dataset, developed by Koroniotis et al. in 2019 [13], is a standard for IoT security research. In this work, we consider the combined training and testing files named:

`UNSW_2018_IoT_Botnet_Final_10_best_Training` and `UNSW_2018_IoT_Botnet_Final_10_best_Testin`

This resulted in approximately *3.67 million records* and 19 features. It contains 11 attack categories such as DDoS, Service Scan, OS Fingerprinting, Data Exfiltration, and Keylogging. It is an attractive set of data because it aims to simulate real botnet attacks on IoT networks. Refer to Table 3 for its full distribution.

Table 3. Class distribution in the Bot-IoT dataset.

Class Label	Count
DoS UDP	1,032,961
DDoS TCP	977,380
DDoS UDP	948,239
DoS TCP	615,800
Reconnaissance Service Scan	64,280
Reconnaissance OS Fingerprint	17,780
DoS HTTP	1,485
DDoS HTTP	989
Normal Normal	477
Theft Keylogging	73
Theft Data Exfiltration	6

ToN-IoT: The ToN-IoT dataset created by Moustafa et al. [14] is an IoT security dataset. In this work, we use only the `train_test_network`-based subset, of the 44-feature set describing *normal* and *malicious* network traffic. The dataset has 10 classes: 9 indicate cyber-attacks like DDoS, DoS, Backdoor, MITM, Injection, Scanning, XSS, Password, and Ransomware, and the remaining one represents normal traffic. The data contains 50,000 normal and 161,043 attack-labeled samples. In Table 4, we present the full class distribution, which makes it a very rich dataset to evaluate intrusion detection systems in real-world IoT settings.

Table 4. Class distribution in the ToN-IoT dataset.

Class Label	Count
normal	50,000
backdoor	20,000
ddos	20,000
dos	20,000
injection	20,000
password	20,000
ransomware	20,000
scanning	20,000
xss	20,000
mitm	1,043

NSL-KDD: The NSL-KDD dataset is a common benchmark for intrusion detection and was first proposed by Tavallaee et al. [16] (2009) as the improved version of the KDD'99 dataset. It recognizes five classes of network traffic-as Normal, DoS, Probe, R2L, and U2R. The dataset comes with a total of 125,973 training records and 22,544 testing records, while 43 features are specified (including the label). Because of its balanced distribution and elimination of redundant instances, this dataset has been used in the literature for evaluating IDS models.

Table 5. Class distribution in the NSL-KDD dataset.

Class	Count
Normal	77,144
DoS	52,986
Probe	13,954
R2L	3,702
U2R	119

We consider from the above that each dataset captures unique IoT threat patterns and offers a varied challenge for robust IDS design.

3.3. Data Preparation

3.3.1. Preprocessing

To feed the proposed Intrusion Detection System (IDS) with regular and quality input to train it, several preprocessing steps are carried out on all four datasets (Edge-IIoTset, Bot-IoT, ToN-IoT, and NSL-KDD). The steps are designed to clean the data, preprocess the categorical features, and normalize the feature space before feeding the data to the deep learning model.

Feature Elimination: Some of the irrelevant or high-cardinality features were dropped from each dataset to eliminate noise and improve model performance. For instance, the Edge-IIoTset dataset had frame.time, ip.src host, and mqtt.topic removed; the Bot-IoT dataset removed pkSeqID, saddr, daddr, and attack; the ToN-IoT dataset removed src ip, http user agent, and weird addl; and the NSL-KDD dataset removed difficulty level. These features were either too specific, too textual, or too redundant, and in all such cases, their removal helped simplify the datasets for more effective training.

Null and Duplicate Handling: All missing (null) values were removed to ensure no invalid entries were passed to the model. Duplicate records were also dropped to eliminate redundancy, which could otherwise bias the model during training.

Encoding: To ready the categorical data for model training, adequate encoding techniques were utilized. One-hot encoding was used on the nominal categorical variables to convert them into binary vectors that could be processed by the model in an efficient manner. In cases where the categorical columns contained large counts of unique values, label encoding was used to encode them using integer labels. These encodings allowed the model to effectively utilize all the non-numerical data.

Once the splitting and encoding processes were completed, each dataset was subjected to an 80-20 split into training and testing sets using stratified sampling to preserve the proportional division of the attack and normal classes. Label encoding was performed next on both training and testing classes to allow for categorical class labels to be used in integer format for classification. Afterward, standard scaling was applied with StandardScaler to give all numerical features a zero mean and unit variance. Normalization helps to stabilize and accelerate the training of deep learning models. These

preprocessing steps were done in all datasets to treat all datasets equally and maintain consistency prior to class balancing and feature selection. It should be noted that the feature counts reported in Section 3.2 correspond to the original datasets prior to preprocessing. After applying one-hot encoding and other preprocessing operations, the dimensionality of the feature space increased for datasets containing categorical attributes. For example, in the NSL-KDD dataset, the feature space expanded from 43 original attributes to 122 encoded features after preprocessing.

3.3.2. Class Balancing

To address class imbalance, the Synthetic Minority Oversampling Technique (SMOTE) was employed. SMOTE generates synthetic instances for minority classes by interpolating between existing samples [27]. This helped achieve a more balanced class distribution and reduced bias toward the dominant class during the training of deep learning models. The successful balancing resulted in the model's enhanced ability to detect attack types that occur less frequently.

In the Edge-IIoTset dataset, SMOTE was applied in smaller batches exclusively to the attack class subset to generate synthetic samples. Following oversampling, the "Normal" class was deliberately under sampled to match the new size of the augmented attack data. This two-step balancing process ensured equitable representation across all classes. Each class—including rare ones like MITM and Ransomware—achieves near-equal sample counts after balancing.

The Bot-IoT, ToN-IoT, and NSL-KDD datasets also had class imbalances, especially with the "Normal" class dominating and some attacks being very rare. This was addressed by applying standard SMOTE to the filtered training data where minority classes were oversampled until they matched the size of the majority class through synthetic sample generation. This method of balancing not only balanced the class distributions but also improved the classifier's detection of rare yet important intrusions.

3.3.3. Feature Selection (Mutual Information)

For dimensionality reduction and for keeping only the most effective features for intrusion classification, Mutual Information (MI) is applied to select the relevant features. The MI score for a feature basically signifies the reduction in uncertainty of the target variable considering the given feature, which is formally expressed with entropy-based metrics. MI checks how dependent each feature is on the target class and is actually able to determine both linear and nonlinear dependencies [28].

It signifies a greater contribution to classification accuracy with higher scores. Moreover, by discarding irrelevant or redundant features, MI-based selection also contributes to model lightweightness, reducing input dimensionality, which is critical for deployment in resource-constrained edge environments. MI scores are computed on the training set prior to any selection, using a non-parametric k-nearest neighbor–based approach. [29]

For each dataset used in this study, after balancing it using SMOTE, MI scores are computed on the training set prior to any selection. Ranked MI scores are plotted to observe the diminishing significance of features and determine the optimal value of k (the number of top features to retain).

The optimal value of k was chosen using empirical analysis of the ranking curve of MI values and determining the elbow (saturation) point after which any further increase in features resulted in only marginal gain in feature importance. Different values near the elbow point were tested, and the one that offered the best balance between classification accuracy and computation was chosen for different datasets.

The reported numbers of selected features correspond to the transformed feature spaces obtained after preprocessing and encoding, rather than the original dataset attributes.

Below or at the elbow point or saturation point in each plot, the following values were selected:

- Edge-IIoTSet: 40 features
- Bot-IoT: 12 features
- ToN-IoT: 35 features
- NSL-KDD: 45 features

These choices are a trade-off between optimizing classifier performance and minimizing computational overhead. By eliminating noisy or redundant features, MI-based selection enhanced both learning efficiency and generalization in all datasets.

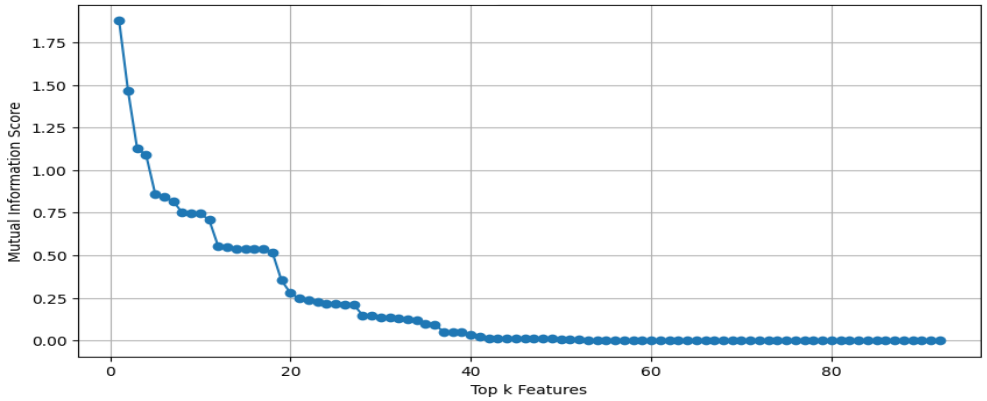


Fig. 2. Mutual Information Score vs. Feature Rank for Edge-IIoTSet dataset

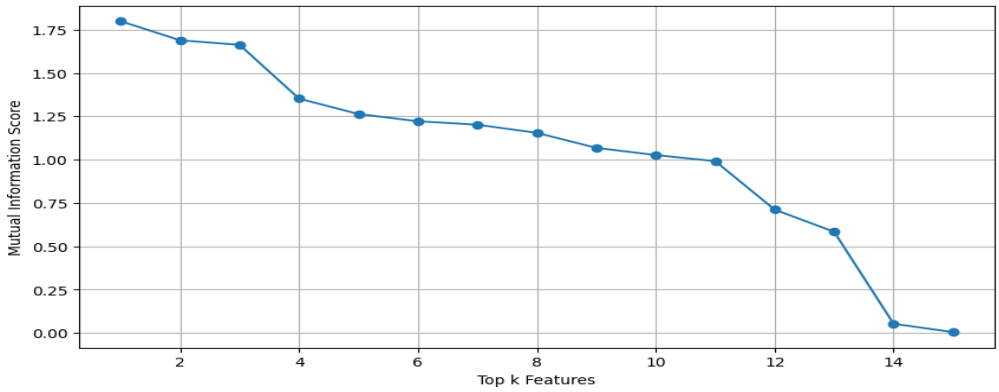


Fig. 3. Mutual Information Score vs. Feature Rank for Bot-IoT dataset.

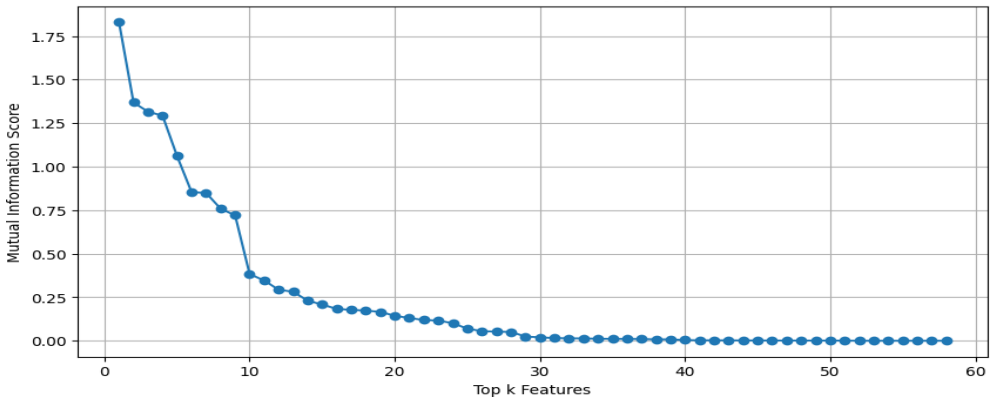


Fig. 4. Mutual Information Score vs. Feature Rank for ToN-IoT dataset.

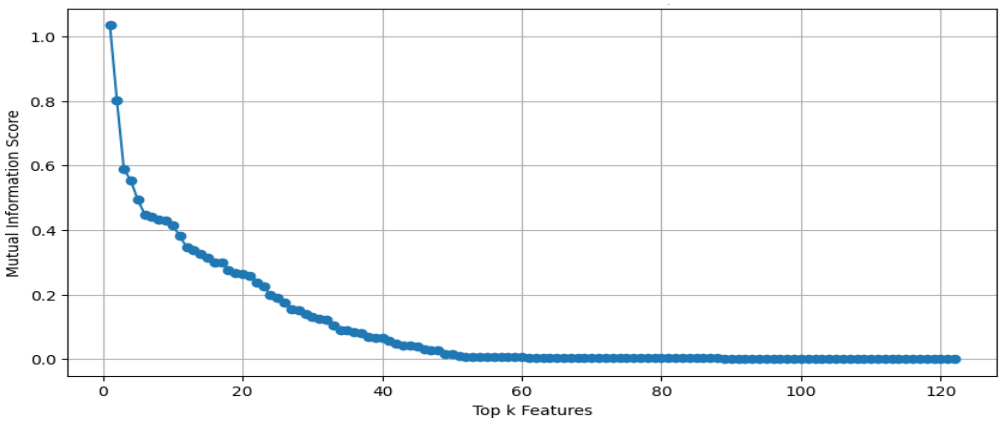


Fig. 5. Mutual Information Score vs. Feature Rank for NSL-KDD dataset.

3.4. Model Architecture

The intrusion detection from network traffic data using a hybrid deep learning model consisting of Convolutional Neural Networks and Gated Recurrent Units is proposed in this study. The hybrid approach tries to take advantage of the CNN in finding spatial patterns in structured input features and that of the GRU in detecting time-based dependencies in sequential data. Through this integration, the model can catch localized anomalies as well as long-range behavior patterns usually seen in cyber-attacks.

CNN: Convolutional Neural Networks (CNNs) effectively extract spatial patterns from structured data, including sequential tabular inputs. In this work, a 1D convolutional layer(Fig. 6) identifies local dependencies and feature correlations within the input network traffic data. The convolution operation generates feature maps by sliding filters over the input sequence. Mathematically, this can be expressed as:

$$D_k = F(W_k * D_{k-1} + \delta_k) \tag{1}$$

Where D_k is the output of the k-th layer, W_k represents the weight matrix, δ_k is the bias, and $*$ denotes the convolution operation, and F acts as an activation function; in this case, the Rectified Linear Unit (ReLU) introduces non-linearity and is defined as:

$$F(x) = \text{Max} \begin{matrix} 0, & \text{for all } x < 0 \\ x, & \text{for all } x \geq 0 \end{matrix} \tag{2}$$

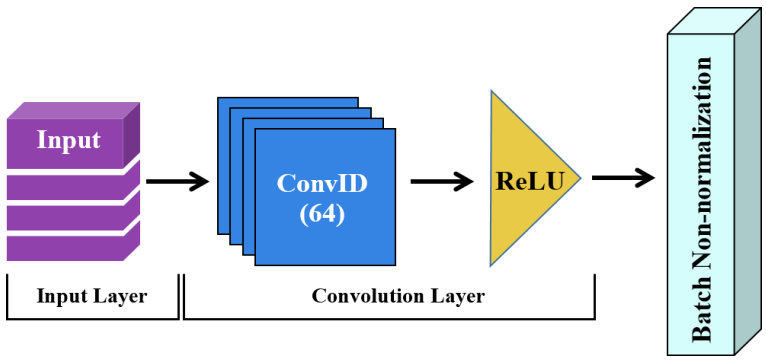


Fig. 6. CNN layer.

This setup allows the model to learn spatial patterns that are important for intrusion detection [30]. Batch normalization follows the convolutional layer to stabilize the learning process, and the GRU component manages temporal dependencies.

GRU: A single Gated Recurrent Unit (GRU) layer (Fig. 7) is used after the convolutional feature extraction block to capture sequential relationships between extracted feature maps. GRUs are a computationally light option compared to RNNs but retain temporal context through gating without the excessive computational cost of standard LSTMs [31].

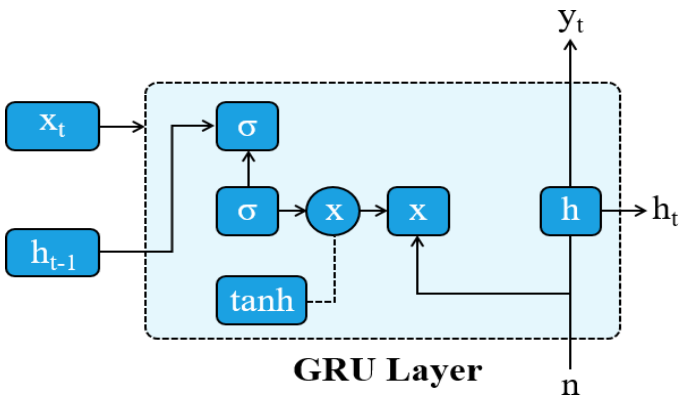


Fig. 7. GRU layer

Mathematically, the GRU operates using two gates: the update gate (z_t) and the reset gate (r_t), defined as:

$$z_t = \sigma(W_z * [h_{t-1}, x_t] + b_z) \quad (3)$$

$$r_t = \sigma(W_r * [h_{t-1}, x_t] + b_r) \quad (4)$$

Where:

- x_t is the input at time t,
- h_{t-1} is the previous hidden state,
- σ *denotes the sigmoid function.*

This process allows the GRU to monitor useful past experience thus improving in distinguishing time-dependent patterns in network traffic data[30].

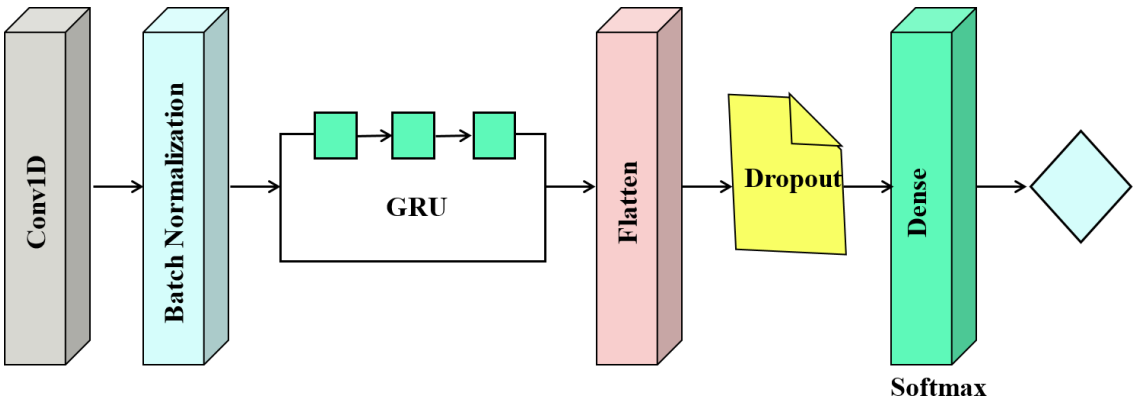


Fig. 8. CNN-GRU Hybrid model.

CNN-GRU Hybrid Model: The CNN-GRU combined method seeks to leverage the strengths of both convolutional and recurrent models in an effort to better and more effectively detect network intrusions within network traffic. The model begins by passing the input features through a convolutional layer, which is best at extracting local spatial pat-terns—feature correlations within short sequences of data. Early detection of key structures or outliers of network activity is crucial in the pipeline. Upon extracting these spatial patterns, the outcome is input to a GRU layer. GRU tracks how these patterns so extracted evolve in time and helps the model to determine whether a sequence of actions suggests normal behavior or a potential attack. After the GRU sequence modeling, the combined information is flattened and input into a stack of dense (fully connected) layers. The latter layers continue to mold the learned features and help with making final discriminations between various attack types. This is facilitated by applying a dropout layer during training in order to prevent overfitting, enhancing the model's generalization on unseen data. The final output layer predicts each class's probability using a softmax function. This makes the model suitable for multi-class classification. The overall structure of the proposed model (Fig. 8) is summarized in Algorithm 1, which outlines each major component of the CNN-GRU pipeline used for intrusion detection. With the integration of CNN and GRU, the model uses the spatial learning and temporal learning and therefore achieves more precise and effective detection of complex and dynamic cyber-attacks in IoT scenarios. Besides, by the usage of feature selection and the optimized reduced model with lower layers and parameters, the model remains lightweight—enabling it to be deployed in real-time on low-resource edge devices.

Algorithm 1 CNN-GRU Hybrid Model **Require:** Input df= dataset
1: Select Features = $f_1, f_2, \ldots, f_n$ from df , where n = no. of selected features
2: Split dffeaturesinto training and testing sets
3: **procedure** MODEL
4: Set inputshape = (T, 1), where T denotes the number of selected features.
5: Add Conv1D Layer (filters = 64, kernelsize = 3, activation= ReLU)
6: Add BatchNormalization layer
7: Add GRU Layer (units = 64, returnsequences = True)
8: Flatten the GRU output
9: Add Dense Layer (units = 64,activation= ReLU)
10:Add Dropout Layer (rate = 0.3)
11: Add Dense Layer (units = 32, activation= ReLU, name = featurevector)

12:Add output Dense layer (units = number of dataset classes, activation = Softmax)
13: Compile the model using the Adam optimizer (learning rate = 0.0001)
14:Set the loss function to Categorical cross-entropy
15:Train the model (batch size = 128) using EarlyStopping
16: **end procedure**

Here, T denotes the number of selected input features after Mutual Information feature selection. Accordingly, T = 40, 12, 35, and 45 for the Edge-IIoTSet, Bot-IoT, ToN-IoT, and NSL-KDD datasets, respectively.

The model was trained using the Adam optimizer with a learning rate of 0.0001 and a batch size of 128. EarlyStopping based on validation loss was employed to prevent overfitting by restoring the best model weights. The maximum number of training epochs was set to 6 for Bot-IoT, 30 for Edge-IIoTSet and NSL-KDD, and 50 for ToN-IoT.

4. Results and Discussions

This section undergoes a complete and detailed testing of the CNN-GRU hybrid intrusion detection model on four benchmark datasets. In the subsections, we discuss the experiment setup, the evaluation metrics, the performance of the model on each dataset, a comparative analysis with some of the state-of-the-art models, and inference time and lightweight analysis.

4.1. Experimental Setup

All experiments are conducted on a personal computer equipped with an Intel Core i7-7700 CPU @ 3.60 GHz, 16 GB RAM, and an NVIDIA GeForce GTX 1070 GPU with 8 GB VRAM, running on a 64-bit Windows 10 operating system. The deep learning models are implemented using Python 3.10 with TensorFlow 2.12 and Keras 2.12.0 in an Anaconda-managed environment. Supporting libraries include scikit-learn 1.3.0, pandas 2.0.3, NumPy 1.24.3, and Matplotlib 3.7.2.

4.2. Evaluation Metrics

To extensively analyze the performance of our proposed model across different datasets, we employ a set of established evaluation metrics, including Accuracy, Precision, Recall, F1-Score, Loss, Training Time, and Inference Time. Each of these metrics serves a specific purpose in measuring the model's correctness, robustness, and computational efficiency.

Accuracy: It is a measure of the overall percentage of samples correctly classified (normal and attacks) among all the samples. It is an overall description of the model's performance across all classes.

$$\text{Accuracy} = \frac{(TP + TN)}{(TP + TN + FP + FN)} \tag{5}$$

Loss: Loss is the training objective minimized during model training. Validation loss, a metric that indicates how well the model generalizes to new, unseen data, is reported here. Improved validation loss values imply improved convergence and reduced error in test dataset predictions.

Precision: It is the measure of accurately predicted positive samples out of all predicted positives. It evaluates the model's ability to avoid false alarms.

$$\text{Precision} = \frac{TP}{TP + FP} \tag{6}$$

Recall: Recall, or sensitivity, measures the number of accurately predicted positive samples out of all genuinely positive samples. It indicates how good the model is at detecting true intrusions.

$$\text{Recall} = \frac{TP}{TP + FN} \tag{7}$$

F1-Score: It is the harmonic mean of precision and recall, giving a balanced view of the model's performance, especially for imbalanced data.

$$\text{F1 Score} = \frac{2 \times \text{Precision} \times \text{Recall}}{\text{Precision} + \text{Recall}} \tag{8}$$

Average Training Time per Epoch: It is a metric of the amount of time the model takes to complete an iteration over the training set. It is a training efficiency measure, which is quite crucial to measure in determining the scalability of the model.

Inference Time: Time taken by the model to infer the output on the test set once the model is trained. In real intrusion detection systems, especially on edge devices, fast inference is extremely crucial.

These metrics together provide a holistic view of model performance across all four datasets. Additionally, graphical evaluation tools such as the *accuracy vs. epoch curve*, *accuracy vs. loss curve*, and the *multiclass confusion matrix* are employed in the Results section to provide a more intuitive and detailed understanding of the model's learning behavior, convergence, and classification performance.

4.3. Performance on Each Dataset

4.3.1. Performance on Edge-IIoT Dataset:

The proposed model has an accuracy of 94.16%, a precision of 96.12%, a recall of 94.15% and an F1-score of 94.02% on the Edge-IIoTSet data set. The loss value obtained at convergence is 0.1064, which confirmed that there is enough optimization without any sign of overfitting. The model demonstrates good generalization capability, as further supported by the strong multiclass ROC-AUC values (Fig. 9.) and the low confusion observed for most classes (Fig. 10.), indicating that the learned representations are discriminative. The multiclass ROC curve (Fig. 9.) also attests to the robustness of the model in terms of displaying AUC values at or very near 1.00 for nearly all classes, implying high sensitivity towards discriminating between attack modes and normal traffic.

The confusion matrix (Fig. 10.) indicates accurate classification for the major classes, i.e. DDoS HTTP, DDoS ICMP, and Uploading, and some minor classes such as Fingerprinting and MITM have negligible misclassification—most likely due to fewer training samples.

The model otherwise provided a fine balance of accuracy and computational cost with a total inference time of 30.59 seconds and an average training time of 90.00 seconds per epoch, thus establishing itself as feasible for resource-constrained IoT environments.

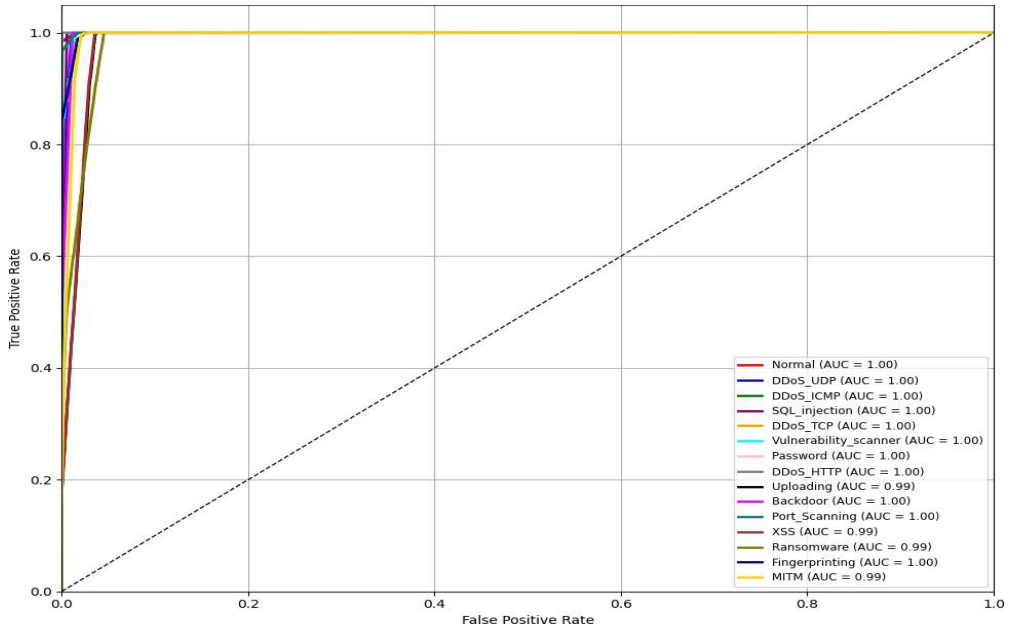


Fig. 9. Multi-Class ROC Curve for the Edge-IIoTSet dataset.



Fig. 10. Multiclass Confusion Matrix for the Edge-IIoTSet dataset(15 classes).

4.3.2. Performance on BoT-IoT Dataset:

The model performed extremely well on the BoT-IoT dataset with an accuracy of 99.31%, precision of 99.32%, recall of 99.30%, and an F1-score of 99.30%. The final loss is negligible at 0.0152, which reflects the successful learning of the model. The ROC curve in Fig. 11. shows an excellent AUC value of 1.00 for all 11 classes, indicating strong classification capability towards identification of different attacks. The confusion matrix in Fig. 12. also shows a correct out-put for important categories of attacks, such as `DDoS TCP`, `DDoS UDP`, and `Reconnaissance OS Fingerprint`. However, there is sometimes confusion arising between comparable attacks, that is, reconnaissance and DoS types, though overall performance is good. At 409.25 seconds per epoch average training time and 46.66 seconds total inference time, the model is still pragmatically efficient for use in resource-limited IoT systems.

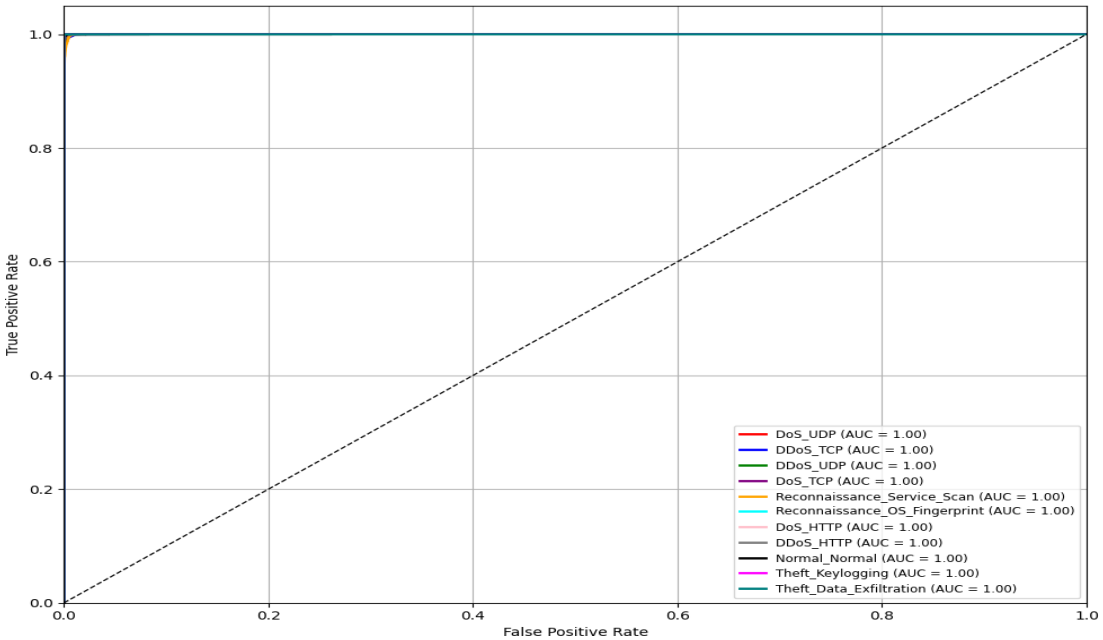


Fig. 11. Multiclass ROC Curve for the BoT-IoT dataset.

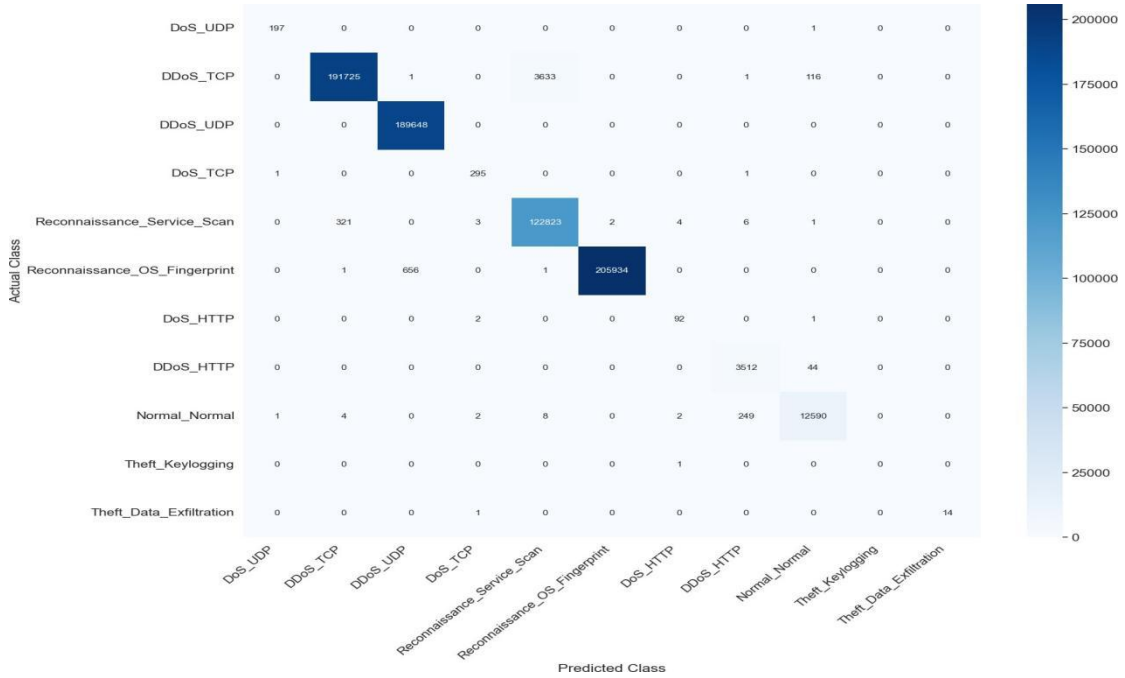


Fig. 12. Multiclass Confusion Matrix for the Edge-IIoTSet dataset(15 classes).

4.3.3. Performance on ToN-IoT Dataset:

The model also performed well on the ToN-IoT dataset, achieving an accuracy of 96.77%, precision of 97.63%, recall of 96.77%, and an F1-score of 97.09%. The final loss is 0.1097, indicating good convergence. The ROC curve (Fig. 13.) displays high AUC values of 1.00, confirming strong class separability. The confusion matrix (Fig. 14.) shows high classification accuracy for major classes like `password`, `ransomware`, and `scanning`, while minor confusion appeared in similar categories such as `dos` and `injection`.

Overall, the model maintained a great balance between accuracy and efficiency, with an average training time per epoch of 21.24 seconds and total inference time of just 3.25 seconds—highlighting its suitability for real-time intrusion detection.

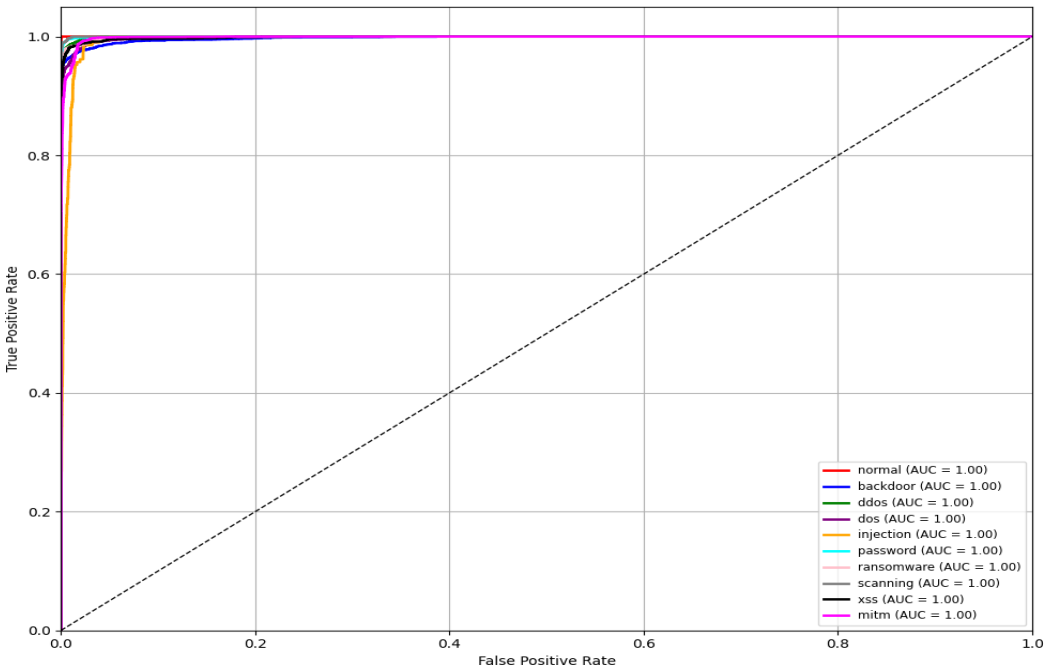


Fig. 12. Multiclass ROC Curve for the ToN-IoT dataset.

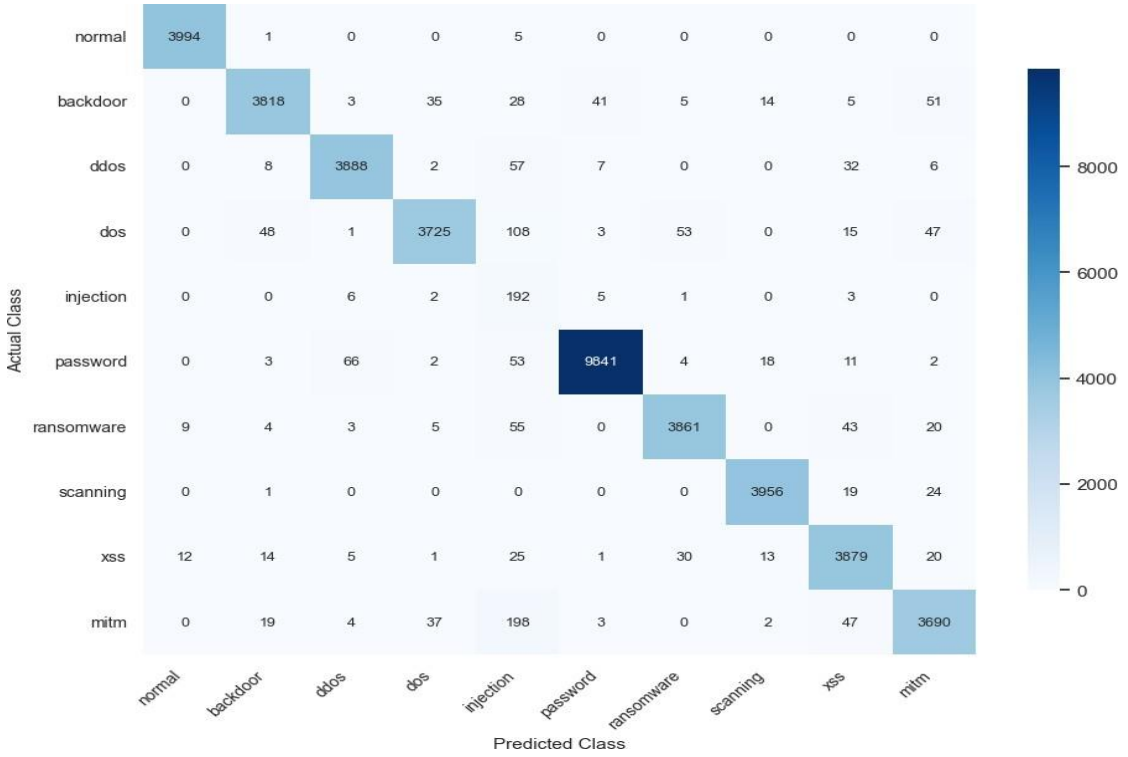


Fig. 13. Multiclass Confusion Matrix for the ToN-IoT dataset.(10 classes)

4.3.4. Performance on NSL-KDD Dataset:

The model shows an accuracy of 98.93%, a precision of 99.02%, a recall of 98.93%, and an F1-score of 98.96% on the NSL-KDD dataset. The model converged quickly with a stopping loss of just 0.0353.

The ROC curves (Fig. 15.) demonstrate high discriminative performance across all five classes, with AUC values 1.00. The confusion matrix (Fig. 16) shows few misclassifications, with great detection in dominant classes such as `Normal` and `DoS`. Generally, the model shows comparatively better performance with only a 17.90-second training time per epoch and only 2.70 seconds of total inference time—demonstrating its capabilities for light IoT-based intrusion detection.

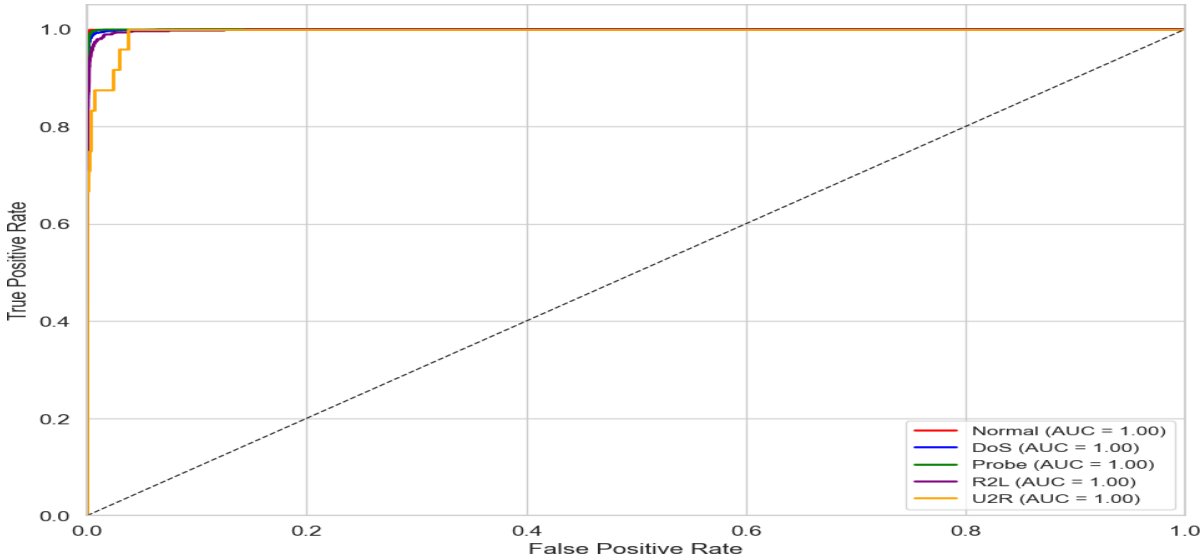


Fig. 14. Multi-class ROC Curve for NSL-KDD dataset.

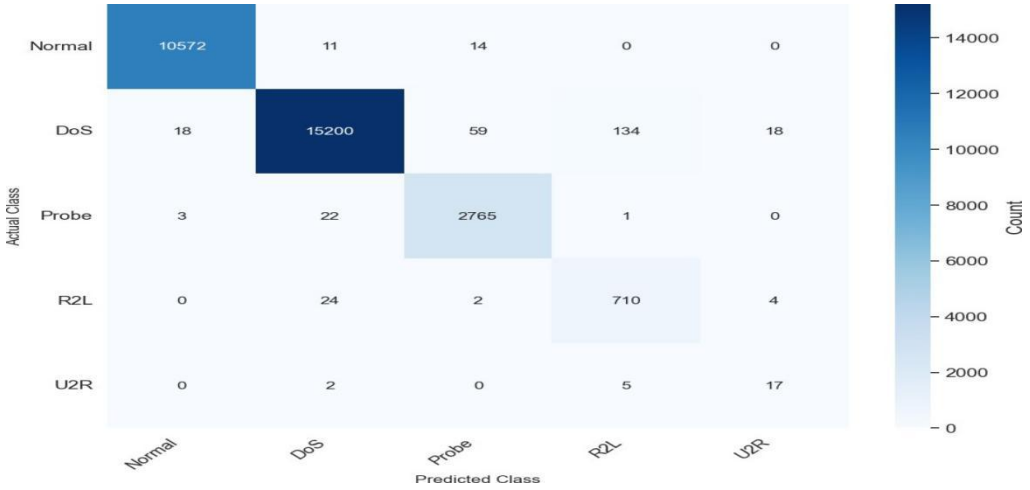


Fig. 15. Multiclass Confusion Matrix for NSL-KDD Dataset (5 Classes).

Table 6 presents the relative performance of the proposed model for all four benchmark datasets. The model achieved the highest accuracy and F1-score for the BoT-IoT dataset (99.30%), which validates its performance for handling highly imbalanced and diverse traffic patterns.

Table 6. Comparative performance of the proposed CNN-GRU model across all datasets

Dataset	Accuracy (%)	Loss (%)	Precision (%)	Recall (%)	F1 Score (%)	Avg Training Time/Epoch(s)	Total Inference Time (s)
Edge-IIoT	94.16	0.1064	96.12	94.15	94.02	90.00	30.59
BoT-IoT	99.31	0.0152	99.32	99.30	99.30	409.25	46.66
ToN-IoT	96.77	0.1097	97.63	96.77	97.09	21.24	3.25
NSL-KDD	98.93	0.0353	99.02	98.93	98.96	17.90	2.70

Fig. 16. shows that on the BoT-IoT (99.31%) dataset, our model achieves better accuracy compared to other tested datasets, although on the NSL-KDD (98.93%) dataset, the model achieves promising performance. In Fig. 17. we show the corresponding inference time per sample (in milliseconds), reflecting the model's computational efficiency. Notably, the model maintains low inference latency across all datasets, with the fastest being 0.064 ms on BoT-IoT.

These results confirm that the model has acceptable generalization across different IoT security datasets and gets a balance between detection accuracy and performance, making it a promising candidate for real-time intrusion detection on resource-constrained IoT environments.

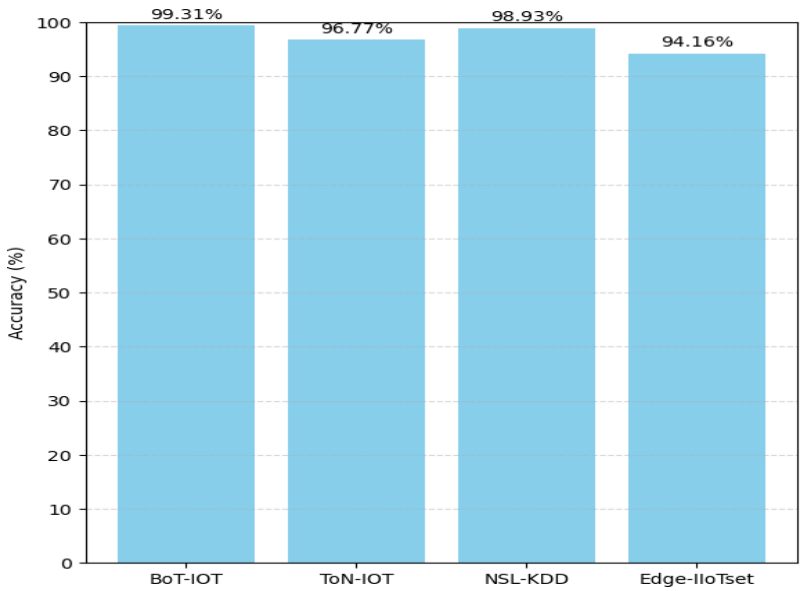


Fig. 16. Model Accuracy across all four Datasets

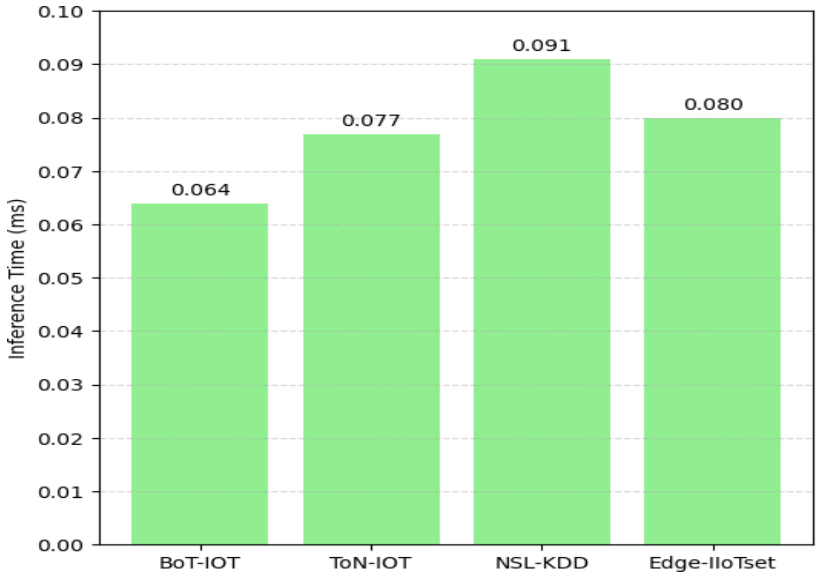


Fig. 17. Model inference time per sample across all four Datasets.

4.4. Comparative Analysis with State-of-the-Art Models

To compare the trade-off in accuracy and computation speed, Table 7 compares the proposed model with various state-of-the-art models on different datasets. The comparison is based on the results reported in the respective publications. Since the hardware platforms, software environments, and training configurations differ across these studies, the comparison should be interpreted as a general performance reference rather than a strictly controlled experimental evaluation. On the Edge-IIoTSet dataset, the DCNN-BiLSTM model [19] achieved a higher accuracy of 99.64%, but it had a significantly higher total inference time of 219.29 seconds and per-sample latency of 0.494 milliseconds. In contrast, the model had a trade-off accuracy of 94.16%, but its reduced overall inference time of 30.59 seconds and ultra-low per-sample latency of only 0.080 milliseconds demonstrate its efficiency in resource-constrained IoT environments where low-latency performance is a priority. On the BoT-IoT dataset, the model achieved an accuracy of 99.31%, while KronNet [11] achieved 99.91%, but with a significantly lower per-sample inference time of 0.064 milliseconds compared to KronNet's 0.208 milliseconds. Although KronNet's total inference time was not reported, the model completed inference on 731,894 test samples in just 46.66 seconds, highlighting its suitability for real-time intrusion detection in resource-limited environments. On the ToN-IoT dataset, the CBCTL-IDS model [32] achieved a high accuracy of 99.85%, but it requires 209 seconds for total inference time, resulting in a per-sample latency of 4.95 milliseconds. The model, with 96.77% accuracy, completed inference in just 3.25 seconds on 42,209 test samples, translating to a per-sample inference time of 0.077 milliseconds, making it more applicable for real-time edge deployment. Finally, on the NSL-KDD dataset, the CNN-GRU model achieved a slightly higher accuracy of 98.93% compared to the IDS-RF model [33], which had an accuracy of 98.54%. More importantly, the model demonstrated superior speed, with a total inference time of 2.70 seconds for 29,581 test samples, or a per-sample latency of 0.091 milliseconds, compared to the IDS-RF model's 7.38 seconds and 0.327 milliseconds per-sample delay. These findings indicate that the proposed CNN-GRU model provides a favorable balance between detection accuracy and low inference latency, making it suitable for resource-constrained IoT environments.

Among the four benchmark datasets, the proposed model achieved better accuracy than the compared model on the NSL-KDD dataset (98.93% vs. 98.54%). In terms of inference time per sample, the model excelled on the Edge-IIoTSet dataset (0.080 ms vs. 0.494 ms), BoT-IoT dataset (0.064 ms vs. 0.208 ms), ToN-IoT dataset (0.077 ms vs. 4.95 ms), and NSL-KDD dataset (0.091 ms vs. 0.327ms).

To provide additional insight into the computational complexity of the proposed CNN-GRU model, the total number of model parameters was also examined. The proposed model contains 77,131, 171,306, 191,951, and 212,101 total parameters for the BoT-IoT, ToN-IoT, Edge-IIoTSet, and NSL-KDD datasets, respectively. Although the parameter count varies naturally with the input dimensionality after feature selection, the overall network architecture remains unchanged. Overall, the proposed model achieved competitive detection accuracy while maintaining low inference latency across all datasets, demonstrating its suitability for real-time intrusion detection in resource-constrained IoT environments.

Table 7. Comparison of proposed model with state of art models across all datasets.

Dataset	Model	Accuracy (%)	Total Inference Time (s)	Inference time per sample (ms)
Edge-IIoTset	DCNN-BiLSTM (Akter et al., 2023) [19]	99.64	219.29	0.494
	Proposed CNN-GRU model	94.16	30.59	0.080
BoT-IoT	KronNet (Ullah et al., 2025) [11]	99.91	-	0.208
	Proposed CNN-GRU model	99.31	46.66	0.064
Ton-IoT	CBCTL-IDS (Zhou et al., 2025) [32]	99.85	209	4.95
	Proposed CNN-GRU model	96.77	3.25	0.077
NSL-KDD	IDS-RF (Alshamy & Akcayol, 2024) [33]	98.54	7.38	0.327
	Proposed CNN-GRU model	98.93	2.70	0.091

4.5. Discussion and Analysis

The outcomes across Edge-IIoTSet, Bot-IoT, ToN-IoT, and NSL-KDD datasets always determine the strength of the proposed CNN-GRU model based on balanced detection accuracy and light deployment loads. The model performs considerably better than others, with low computational resources and stable convergence behaviors, making it a strong candidate for real-time intrusion detection. Notably, the model demonstrated optimal F1-scores and accuracy on Bot-IoT and NSL-KDD with minimal loss in accuracy on more complex datasets such as ToN-IoT and Edge-IIoTSet. But it shows more efficient inference time. For instance, when making inference on ToN-IoT, the model made inference on over 42,209 samples in just 3.25 seconds, while the CBCTL-IDS baseline [32] took 209 seconds. Therefore, in each dataset, latency on a sample-to-sample basis is less than 0.091 milliseconds—a testament to the real-time capability of the model. Another key observation is the robustness of the model's generalization. Training vs. validation curves, ROC-AUC curves and confusion matrices demonstrate virtually no overfitting and discriminative ability in multi-class situations with unbalanced classes. Although a few misclassifications were observed for minority attack classes such as MITM (Edge-IIoTSet) and between closely related categories such as Reconnaissance and DoS (BoT-IoT) and DoS and Injection (ToN-IoT), these are mainly due to the limited number of training samples and the similarity of traffic patterns among certain attack types. Nevertheless, the confusion matrices show that such errors were infrequent and did not significantly affect the overall detection performance. This indicates that the proposed model remains reliable for multiclass intrusion detection across different IoT datasets. Such robustness is particularly important in chaotic and heterogeneous IoT traffic environments where traditional models may degrade.

5. Conclusion

In this article, we propose a CNN-GRU-based efficient and lightweight hybrid deep learning model for an Intrusion Detection System for resource-constrained IoT devices. The model is capable of capturing the temporal and spatial characteristics of traffic patterns. In addition, feature selection using mutual information and SMOTE dataset balancing led to better input representation and improved learning on unbalanced datasets. The model was thoroughly evaluated on four IoT security benchmark datasets, where it consistently reported low per-sample inference times. Despite modest compromises in accuracy on some datasets, the CNN-GRU model showcased lightweight operations and minimum computational overhead. Overall, the proposed CNN-GRU framework offers an efficient balance between detection and computation capability. The proposed CNN-GRU framework is therefore well suited for intrusion detection in IoT systems that have limited resources. While the proposed model performed well in benchmark data sets, its effectiveness in real-world IoT systems has yet to be validated. In future work, we will deploy the model in real-world IoT environments and investigate adaptive learning techniques to improve its robustness against evolving cyber threats.

All the Declarations and Statements

Author Contributions Statement
Safwan Ishrak: Conceptualization, Methodology design and framework development used, Dataset preprocessing, Model Training, Formal analysis, Visualization, Writing original draft.
Puja Dhar: Conceptualization, writing original draft, Review and editing.
Md.Abdul Wahab: Writing, Editing, Academic guidance.
A R M Mahamudul Hasan: Proposed research ideas, Supervision, Methodology, Project administration, Validation, Writing, Review and editing.
Ratnadip Kuri: Investigation, Writing, Review and editing, Validation, Academic guidance
Dr. Mohammed Humayun Kabir: Review and editing, Academic guidance.

All authors have read and agreed to the published version of the manuscript.

Conflict of Interest Statement
The authors declare no conflicts of interest.

Funding Declaration
This study received no external funding from any public, commercial, or not-for-profit funding agencies.

Data Availability Statement
The datasets used in this study are publicly available. We used four benchmark intrusion detection datasets: Edge-IIoTset [15], Bot-IoT [13], ToN-IoT [14], and NSL-KDD [16]. These datasets are widely used for evaluating intrusion detection systems and are available from their respective public repositories.

Ethical Declarations
This study used only publicly available benchmark intrusion detection datasets. No human subjects were directly involved in the data collection and no experimental intervention was conducted.

Acknowledgments
We sincerely thank the reviewer and editor for their professional evaluation and valuable recommendations, which contributed to improving the quality of the experiment and reliability of its results.

Declaration of Generative AI in Scholarly Writing
During the preparation of this manuscript, generative AI tools were used for language refinement and formatting. All research ideas, methodology, experiments, analysis, and conclusions were developed solely by the authors, who take full responsibility for the content of this manuscript.

Abbreviations
The following abbreviations are used in this manuscript:

Abbreviation	Full Form
AI	Artificial Intelligence
AUC	Area Under the Curve
Bot-IoT	Botnet Internet of Things Dataset
CNN	Convolutional Neural Network
DDoS	Distributed Denial-of-Service
DoS	Denial-of-Service
DNN	Deep Neural Network
Edge-IIoTset	Edge Industrial Internet of Things Dataset
GNN	Graph Neural Network
GRU	Gated Recurrent Unit
GPU	Graphics Processing Unit
IDS	Intrusion Detection System
IoT	Internet of Things
MITM	Man-in-the-Middle Attack
MI	Mutual Information
ML	Machine Learning
NSL-KDD	NSL-KDD Intrusion Detection Dataset

ReLU	Rectified Linear Unit
RNN	Recurrent Neural Network
ROC	Receiver Operating Characteristic
SMOTE	Synthetic Minority Over-sampling Technique
ToN-IoT	Telemetry of Network and Internet of Things

Appendix A/B/C…, with appendix title

Not applicable.

References

[1] L. Thomas, B. Anoop, An Efficient IoT Based Intrusion Detection System Using Optimization Kernel Extreme Learning Machine. *International Journal of Computer Network and Information Security 17*, 2025. https://doi.org/10.5815/ijcnis.2025.02.05

[2] M. Fatima, O. Rehman, I.M. Rahman, A. Ajmal, S.J. Park, Towards ensemble feature selection for lightweight intrusion detection in resource-constrained IoT devices. Future Internet. 2024 Oct 12;16(10):368. https://doi.org/10.3390/fi16100368

[3] L. Atzori, A. Iera, G. Morabito, The internet of things: A survey, Computer Networks 54.15 (2010) 2787–2805. https://doi.org/10.1016/j.comnet.2010.05.010

[4] G.A. Mukhaini, M. Anbar, S. Manickam, T.A. Al-Amiedy, A.A. Momani, A systematic literature review of recent lightweight detection approaches leveraging machine and deep learning mechanisms in Internet of Things networks, Journal of King Saud University Computer and Information Sciences 36.1(2024) 101866. https://doi.org/10.1016/j.jksuci.2023.101866

[5] D. Gaikwad, Intrusion detection system using ensemble of rule learners and first search algorithm as feature selectors, International Journal of Computer Network and Information Security (IJCNIS) 13.4 (2021) 26–34. https://doi.org/10.5815/ijcnis.2021.04.03

[6] C. Yin, Y. Zhu, J. Fei, X. He, A deep learning approach for intrusion detection using recurrent neural networks, Ieee Access 5 (2017) 21954–21961. https://doi.org/10.1109/ACCESS.2017.2762418

[7] S. Zavrak, M. Iskefiyeli, Anomaly-based intrusion detection from network flow features using variational autoencoder, IEEe Access 8 (2020) 108346–108358. https://doi.org/10.1109/ACCESS.2020.3001350

[8] V. Hnamte, H. Nhung-Nguyen, J. Hussain, Y. Hwa-Kim, A novel two-stage deep learning model for network intrusion detection: LSTM-AE, Ieee Access. 11 (2023) .https://doi.org/10.1109/ACCESS.2023.3266979

[9] B. Cao, C. Li, Y. Song, Y. Qin, C. Chen, Network intrusion detection model based on CNN and GRU, Applied Sciences 12.9 (2022) 4184. https://doi.org/10.3390/app12094184

[10] M. Jouhari, M. Guizani, Lightweight CNN-BiLSTM based intrusion detection systems for resource-constrained IoT devices, in: 2024 International Wireless Communications and Mobile Computing (IWCMC), IEEE, 2024: pp. 1558–1563. https://doi.org/10.1109/IWCMC61514.2024.10592352

[11] S. Ullah, J. Wu, M.M. Kamal, A.K.J. Saudagar, KronNet a lightweight Kronecker enhanced feed forward neural network for efficient IoT intrusion detection, Scientific Reports 15 .1 (2025) 20850. https://doi.org/10.1038/s41598-025-08921-3

[12] L.K.G. Danquah, S.Y. Appiah, V.A. Mantey, I. Danlard, E.K. Akowuah, Computationally efficient deep federated learning with optimized feature selection for iot botnet attack detection, Intelligent Systems with Applications 25 (2025) 200462. https://doi.org/10.1016/j.iswa.2024.200462

[13] N. Koroniotis, N. Moustafa, E. Sitnikova, B. Turnbull, Towards the development of realistic botnet dataset in the internet of things for network forensic analytics: Bot-iot dataset, Future Generation Computer Systems 100 (2019) 779–796. https://doi.org/10.1016/j.future.2019.05.041

[14] N. Moustafa, A new distributed architecture for evaluating AI-based security systems at the edge: Network TON_IoT datasets, Sustainable Cities and Society 72 (2021) 102994. https://doi.org/10.1016/j.scs.2021.102994

[15] M.A. Ferrag, O. Friha, D. Hamouda, L. Maglaras, H. Janicke, Edge-IIoTset: A new comprehensive realistic cyber security dataset of IoT and IIoT applications for centralized and federated learning, IEEe Access 10 (2022) 40281–40306. https://doi.org/10.1109/ACCESS.2022.3165809

[16] M. Tavallaee, E. Bagheri, W. Lu, A.A. Ghorbani, A detailed analysis of the KDD CUP 99 data set, in: 2009 IEEE Symposium on Computational Intelligence for Security and Defense Applications, Ieee, 2009: pp. 1–6. https://doi.org/10.1109/CISDA.2009.5356528

[17] K. Mounika, P.V. Rao, A. Anbalagan, Modified CNN Model for Network Intrusion Detection and Classification System Using Local Outlier Factor-based Recursive Feature Elimination, International Journal of Computer Network and Information Security 17.1 (2025) 82–91. https://doi.org/10.5815/ijcnis.2025.01.07

[18] J. Sinha, M. Manollas, Efficient deep CNN-BiLSTM model for network intrusion detection, in: Proceedings of the 2020 3rd International Conference on Artificial Intelligence and Pattern Recognition, 2020: pp. 223–231. https://doi.org/10.1145/3430199.3430224

[19] V. Hnamte, J. Hussain, DCNNBiLSTM: An efficient hybrid deep learning-based intrusion detection system, Telematics and Informatics Reports 10 (2023) 100053. https://doi.org/10.1016/j.teler.2023.100053

[20] M. Catillo, A. Pecchia, U. Villano, A deep learning method for lightweight and cross-device IoT botnet detection, Applied Sciences 13.2 (2023) 837. https://doi.org/10.3390/app13020837

[21] S.I. Popoola, B. Adebisi, M. Hammoudeh, G. Gui, H. Gacanin, Hybrid deep learning for botnet attack detection in the internet-of-things networks, IEEE Internet of Things Journal 8.6 (2021) 4944–4956. https://doi.org/10.1109/JIOT.2020.3034156

[22] F. Zawaideh, G. Al-Asad, G. Swaneh, S. Batainah, H. Bakkar, Intrusion detection system for (IoT) networks using Convolutional Neural Network (CNN) and XGBoost algorithm 102(2024). https://www.researchgate.net/publication/385591170_Intrusion_Detection_System_for_IoT_Networks_Using_Convolutional_Neural_Network_CNN_and_XGboost_Algorithm.
[23] A. Qaddos, M.U. Yaseen, A.S. Al-Shamayleh, M. Imran, A. Akhunzada, S.Z. Alharthi, A novel intrusion detection framework for optimizing IoT security, Scientific Reports 14.1(2024) 21789. https://doi.org/10.1038/s41598-024-72049-z
[24] M. Jouhari, H. Benaddi, K. Ibrahimi, Efficient intrusion detection: Combining x 2 feature selection with CNN-BiLSTM on the UNSW-NB15 dataset, in: 2024 11th International Conference on Wireless Networks and Mobile Communications (WINCOM), IEEE, 2024: pp. 1–6. https://doi.org/10.1109/WINCOM62286.2024.10658099
[25] T. Altaf, X. Wang, W. Ni, G. Yu, R.P. Liu, R. Braun, GNN-based network traffic analysis for the detection of sequential attacks in IoT, Electronics 13.12 (2024) 2274. https://doi.org/10.3390/electronics13122274
[26] J.P. Singh, R. Kazmi, Fusion Sec-IoT: A Federated Learning-Based Intrusion Detection System for Enhancing Security in IoT Networks., International Journal of Advanced Computer Science & Applications 15.11 (2024). https://doi.org/10.14569/IJACSA.2024.0151116
[27] N.V. Chawla, K.W. Bowyer, L.O. Hall, W.P. Kegelmeyer, SMOTE: synthetic minority over-sampling technique, Journal of Artificial Intelligence Research 16 (2002) 321–357. https://doi.org/10.1613/jair.953
[28] H. Peng, F. Long, C. Ding, Feature selection based on mutual information criteria of max-dependency, max-relevance, and min-redundancy, IEEE Transactions on Pattern Analysis and Machine Intelligence 27.8 (2005) 1226–1238. https://doi.org/10.1109/TPAMI.2005.159
[29] A. Kraskov, H. Stögbauer, P. Grassberger, Estimating mutual information, Physical Review E—Statistical, Nonlinear, and Soft Matter Physics 69.6 (2004) 066138. https://doi.org/10.1103/PhysRevE.69.066138
[30] Y. Imrana, Y. Xiang, L. Ali, A. Noor, K. Sarpong, M.A. Abdullah, CNN-GRU-FF: a double-layer feature fusion-based network intrusion detection system using convolutional neural network and gated recurrent units, Complex & Intelligent Systems 10.3(2024) 3353–3370. https://doi.org/10.1007/s40747-023-01313-y
[31] J. Chung, C. Gulcehre, K. Cho, Y. Bengio, Empirical evaluation of gated recurrent neural networks on sequence modeling, arXiv Preprint arXiv:1412.3555 (2014). https://doi.org/10.48550/arXiv.1412.3555
[32] H. Zhou, H. Zou, P. Zhou, Y. Shen, D. Li, W. Li, CBCTL-IDS: A transfer learning-based intrusion detection system optimized with the black kite algorithm for IoT-enabled smart agriculture, IEEE Access 13 (2025) 46601–46615. https://doi.org/10.1109/ACCESS.2025.3550800
[33] R. Alshamy, M. AKCAYOL, Intrusion detection model using machine learning algorithms on NSL-KDD dataset, International Journal of Computer Networks and Communications 16.6(2024) https://doi.org/10.5121/ijcnc.2024.16605.

Authors' Profiles



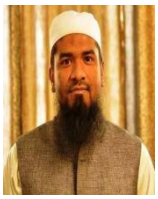
Safwan Ishrak received his B.Sc. (Engineering) in Computer Science and Telecommunication Engineering from Noakhali Science and Technology University (NSTU), Bangladesh. His research interests include software development, machine learning, deep learning, and network security. He is currently engaged in research on intrusion detection systems using hybrid deep learning approaches.



Puja Dhar received her B.Sc. (Engineering) in Computer Science and Telecommunication Engineering from Noakhali Science and Technology University (NSTU), Bangladesh. Her research focuses on designing lightweight and efficient deep learning models for enhancing the security of IoT networks. She has specifically explored the integration of convolutional and recurrent neural architectures with feature selection techniques to develop resource-aware intrusion detection solutions.



Md.Abdul Wahab is serving as a Lecturer at BGC Trust University Bangladesh in Department of Computer Science and Engineering. He has completed his B.Sc. (Engr.) in Computer Science & Telecommunication Engineering (CSTE) from Noakhali Science & Technology University and pursuing his M.Sc. in CSE He has also completed his MBA in Accounting from Chittagong University Center for Business Administration (CUCBA). He is interested to research and work in different areas like Artificial intelligence, Machine learning, Deep learning, Computer Vision, Natural language processing, Cloud computing and Cyber security.



ARM Mahamudul Hasan received his B.Sc. (Honors) degree in Computer Science and Telecommunication Engineering from Noakhali Science and Technology University (NSTU), Bangladesh and his M.Sc. degree from South Asian University (SAARC University), New Delhi, India. He is currently working as an Assistant Professor in the Department of Computer Science and Telecommunication Engineering at NSTU, Noakhali-3814, Bangladesh. His research interests include intrusion detection systems, attack analysis (DoS, DDoS, and emerging cyber threats), IoT and Edge–IoT security, machine learning, deep learning, transfer learning, network security, and blockchain technology. He actively teaches courses on algorithm design & analysis, Structured programming, data communications and cryptography. He is dedicated to advancing research in AI-driven cyber security and intelligent systems.



Ratnadip Kuri is an Assistant Professor in the Department of Computer Science and Telecommunication Engineering at Noakhali Science and Technology University, Bangladesh. He holds both Bachelor's and Master's degrees in Computer Science and Telecommunication Engineering. His research interests include Network Security, Intrusion Detection Systems, Machine Learning, Natural Language Processing, and Computer Vision. He has contributed to multiple research projects across these domains and is actively involved in supervising undergraduate research. Mr. Kuri is also engaged in academic curriculum development aimed at strengthening applied computing education.



Dr. Mohammed Humayun Kabir received B.Sc. (Honors) and M.Sc. degrees in 1993 and 1995 respectively from the Department of Applied Physics and Electronics, the University of Dhaka, Bangladesh. He got Ph.D. in system engineering from the department of Electrical and Electronic Engineering, Kitami Institute of Technology, Hokkaido, Japan. He was a Lecturer and an Assistant Professor in Computer Science and Information Technology, The University of Comilla, Bangladesh. He is working as a Professor in Computer Science and Telecommunication Engineering at Noakhali Science and Technology University, Noakhali-3814, Bangladesh. His research work concerned about Power System Engineering. Now, he is working on Machine Learning and on security issues in Computer Networks.

How to cite this paper

Safwan Ishrak, Puja Dhar, Md. Abdul Wahab, Ratnadip Kuri, ARM Mahamudul Hasan Rana, Humayun Kabir, "A Lightweight CNN-GRU Based Intrusion Detection Model with Mutual Information Feature Selection for IoT Edge Devices", International Journal of Wireless and Microwave Technologies(IJWMT), Vol.16, No.5, pp. 66-88, 2026. DOI:10.5815/ijwmt.2026.05.05