I.J. Wireless and Microwave Technologies, **2025, 5, 57-67**
Published Online on October 8, 2025 by MECS Press (http://www.mecs-press.org/)
DOI: 10.5815/ijwmt.2025.05.05

Cyber Guard: Detecting DoS and DDoS Attack

Jinsu Anna John*
Dept. of Computer Science and Engineering (Cyber Security), India
Email: jinsuanna23@gmail.com
ORCID iD: https://orcid.org//0009-0005-1945-5309
*Corresponding Author

Raj Kumar T.
Dept. of Computer Science and Engineering (Cyber Security), India
Email: rajkumar@cek.ac.in
ORCID iD: https://orcid.org//0009-0001-7558-0076

Shilpa Harrison
Dept. of Computer Science and Engineering (Cyber Security), India
Email: shilpaharrison20@gmail.com
ORCID iD: https://orcid.org//0009-0001-1298-0806

Received: 11 April, 2025; Revised: 26 May, 2025; Accepted: 25 July, 2025; Published: 08 October, 2025

Abstract: The growing adoption of Internet of Things (IoT) devices has amplified the need for robust security mechanisms, particularly against Denial of Service (DoS) and Distributed Denial of Service (DDoS) attacks. This paper proposes a deep learning-based detection system using a hybrid Convolutional Neural Network–Gated Recurrent Unit (CNN-GRU) model to effectively capture both spatial and temporal patterns of malicious activity. The CICDDoS2019 dataset is employed for training and evaluation, with preprocessing steps including Boruta-based feature selection and data rebalancing using SMOTE. A user-friendly GUI developed in Python (Tkinter) facilitates real-time input and prediction. The proposed model, Cyber Guard, demonstrates high accuracy and efficiency, offering a practical solution for IoT attack detection and future deployment.

Index Terms: IoT, DoS, DDoS, CNN GRU

1. Introduction

In recent years, the widespread adoption of Internet of Things (IoT) devices has brought about unprecedented connectivity and convenience across various sectors, spanning from smart homes to industrial automation. Nevertheless, the extensive integration of IoT technologies has also exposed a new realm of vulnerabilities, particularly concerning the security of these interconnected devices [1]. A primary challenge facing IoT ecosystems is their susceptibility to malicious activities, notably Denial of Service (DoS) and Distributed Denial of Service (DDoS) attacks. IoT devices, known for their diversity and resource limitations, present a complex security landscape. Traditional security measures often prove insufficient in safeguarding these devices against the sophisticated and evolving nature of cyber threats. DoS and DDoS attacks, which aim to disrupt or overwhelm the normal operation of IoT devices or networks, pose a significant risk to the reliability and integrity of IoT-enabled systems.

In a DoS attack, the attacker floods a device or network with so many requests that legitimate users are unable to access the intended service [2]. DDoS attacks exacerbate the impact by coor- dinating multiple compromised devices to mount a concerted assault on a target. In response to these challenges, the primary focus of this project is on developing a robust detection mechanism for DoS and DDoS attacks targeting IoT systems.

To enhance the detection of malicious behavior in IoT networks, this study employs a hybrid deep learning model that combines Convolutional Neural Networks (CNN) and Gated Recurrent Units (GRU). CNNs are leveraged to capture spatial patterns from the input data, while GRUs are used to model temporal dependencies, making the approach well-suited for analyzing complex, time-varying traffic patterns typical of IoT environments. By integrating these two architectures, the system is capable of identifying subtle, context-specific anomalies that may indicate ongoing attacks. As IoT networks expand and diversify, the need for adaptive and intelligent security frameworks becomes more pressing. The outcomes of this research not only enhance the defense mechanisms for IoT systems but also contribute valuable insights into the application of deep learning techniques in cybersecurity

While CNN-GRU models have been applied in other domains such as video recognition and natural language processing, their adoption in IoT security, particularly for detecting a wide range of reflective DoS and DDoS attacks, remains underexplored. This work uniquely leverages the strengths of both CNN and GRU to capture spatial-temporal attack patterns specific to IoT traffic, which is often sparse, imbalanced, and multi-modal. Additionally, the integration of Boruta-based feature selection and SMOTE-driven balancing enhances the model's robustness and generalization across diverse attack classes. The development of a practical, real-time GUI tool further distinguishes this work by offering deployability and user accessibility, which is rarely addressed in similar research

2. Literature Survey

Hussain et al. [3] propose a deep learning-based method for identifying sophisticated DoS and DDoS attacks in IoT networks using the ResNet architecture. Their study demonstrates how the integration of deep neural networks with traditional security approaches can significantly improve threat detection capabilities within IoT systems.

Kajwadkar and Jain [4] present a novel algorithm aimed at the early detection and mitigation of DoS and DDoS attacks in IoT environments. Their work highlights the importance of protecting resource-constrained IoT networks and demonstrates that the proposed approach outperforms existing techniques. The study also suggests the potential for future improvements and deeper analysis.

In their work, Kumar and Kulothungan [5] focus on DoS attacks specifically targeting IoT systems. They introduce a Topology Management Method (TMM) for attack prevention, com- bined with a Fine-Grained Detection Algorithm aimed at identifying anomalies in the MAC layer protocols. The study highlights the importance of behavioral analysis and presents a modular framework for both detecting and mitigating DoS threats in IoT environments.

Liang et al. [6] examine various Denial of Service (DoS) attack strategies targeting IoT systems using the Kali Linux platform. Their study evaluates and compares the effectiveness of three existing DoS techniques and also proposes a new method specifically designed for IoT environments. The primary goal is to investigate diverse attack mechanisms and assess their impact to strengthen IoT security measures.

Nandi, Phadikar, and Majumder [7] focus on detecting DDoS attacks in cloud-based en- vironments. Their study employs machine learning classifiers combined with hybrid feature selection techniques to identify and categorize malicious network packets. The proposed method is thoroughly evaluated using cross-validation and experimental analysis to assess its performance and reliability.

Soe, Santosa, and Hartanto [8] focus their research on detecting DDoS attacks within IoT environments. They propose a detection framework that utilizes a Simple Artificial Neural Network (ANN) in conjunction with the Synthetic Minority Over-sampling Technique (SMOTE) to manage class imbalance in the dataset. Their approach is designed to improve both the accuracy and reliability of DDoS detection in IoT-based security systems.

The survey conducted by Sonar and Upadhyay [9] provides a comprehensive overview of DDoS attacks targeting IoT systems. It classifies different attack types based on the layers of the IoT architecture and underscores the importance of implementing effective detection strategies. The study also examines multiple attack scenarios and suggests protective measures to strengthen IoT infrastructure against DDoS threats.

The study by Wankhede and Kshirsagar [10] investigates the detection of application-layer DoS attacks using machine learning and neural network techniques. Their work compares the performance of Random Forest (RF) and Multi-Layer Perceptron (MLP), concluding that RF provides better classification accuracy. The methodology includes data preprocessing and training on the CICIDS 2017 dataset, followed by applying RF and MLP models for identifying DoS traffic patterns.

3. Proposed Methodology

The four main components of the suggested methodology are Attack pattern Recognition, Data Conversion, Data preprocessing, and Data Acquisition.

The initially suggested methodology involves choosing “CICDDoS2019”, which is accessible to the public. A collection of benign and current popular DDoS attacks that mimic actual real-world data (PCAPs) is called CICDoS2019. The results of network traffic analysis using CICFlowMeter-V3, with flows labelled based on date, source and destination IPs, source and destination ports, protocols, and attack, are included in this version, though, and it is in CSV file format. We have a variety of contemporary reflective DDoS assaults in this dataset, including ones using PortMap, NetBIOS, LDAP, MSSQL, UDP, UDP-Lag, SYN, NTP, DNS, and SNMP. Subsequent attacks were carried out throughout this time.On the training day, we carried out 12 DoS and DDoS assaults, including those against NTP, DNS, LDAP, MSSQL, NetBIOS, SNMP, SSDP, UDP, UDP-Lag, WebDDoS, SYN, and TFTP.On the testing day, there were seven attacks: PortScan, NetBIOS, LDAP, MSSQL, UDP, UDP-Lag, and SYN. WebDDoS traffic volume was very low, therefore PortScan was only used on the testing day and will not be used to assess the suggested model.

A total of 11 classes were identified using the Cyber_Guard model. Class 0: Benign, Class 1: DrDoS_DNS, Class 2: DrDoS_NTP, Class 3: DrDoS_SNMP, Class 4: LDAP, Class 5: MSSQL, Class 6: NetBIOS, Class 7: Syn, Class 8: TFTP, Class 9: UDP, Class 10: UDPLag, Class 11: WebDDoS

Fig 1 represents Data Distribution Plot which shows the distribution of classes (attack types) in the dataset to visualize class imbalance.

The preprocessing phase is a critical step conducted after acquiring the raw data, aimed at refining it for efficient and accurate model training. This phase involves two key processes: cleaning the dataset and converting the refined data into a format compatible with CNN input—specifically, transforming the data into structured, image-like arrays suitable for deep learning.

The process begins with loading the CICDDoS2019 dataset in CSV format. All rows containing missing or null values are detected and removed, as their proportion is negligible and does not justify imputation. Following this, non-informative or redundant features—such as IP addresses, port numbers, timestamps, and protocol names—are eliminated since they do not contribute meaningfully to attack detection and may introduce unnecessary noise.

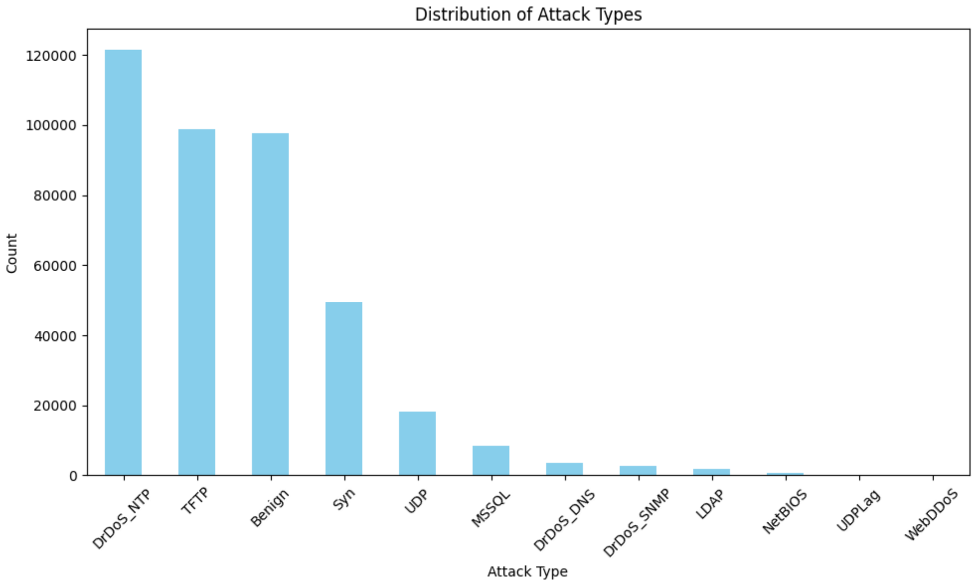


Fig. 1. Distribution of Attack Types

Emphasis is placed on retaining only numerical attributes that capture essential aspects of network traffic behavior, such as flow duration, packet size statistics, and inter-arrival times. As most of these features are already scaled, additional normalization is not required. Categorical variables, if present, are encoded using label encoding to ensure compatibility with the model. By performing this rigorous preprocessing, the dataset is transformed into a clean, structured, and relevant input format, enabling the CNN-GRU model to effectively learn distinguishing patterns and accurately classify DoS and DDoS attacks.

In the final phase of the system, the CNN model is trained and evaluated using the preprocessed dataset to assess its capability in accurately identifying DoS and DDoS attack patterns. Once the model demonstrates satisfactory performance, a user-friendly graphical user interface (GUI) is developed using Python’s Tkinter library to simplify user interaction. The GUI is designed as a single-page application, featuring an input panel for uploading test data and a corresponding output panel that displays the predicted results—such as the attack type—on the same screen.

The preprocessing pipeline begins with loading the dataset, followed by addressing any missing values to maintain data quality. Unnecessary columns that do not contribute to model learning are removed to streamline the dataset. After these cleaning steps, the data is split into training and testing subsets to enable a reliable evaluation of the Cyber_Guard model’s generalization performance on previously unseen traffic data. This comprehensive workflow ensures both the technical robustness of the model and the accessibility of the system through an interactive GUI.

Feature selection is then initiated using machine learning methods, aiming to determine which features are most important for training the model. Data balancing techniques such as the SMOTE algorithm [3] are applied to address class imbalance issues if present. Although SMOTE is widely used to handle class imbalance by producing synthetic data points, it can sometimes lead to overfitting—especially when applied to small datasets or those with high-dimensional features. To minimize this risk in our approach, we first applied the Boruta feature selection algorithm [13]. This step reduced the feature space and ensured that only the most informative attributes were retained, thereby avoiding the generation of synthetic samples in irrelevant or sparse regions of the dataset. Furthermore, we employed cross-validation along with thorough testing on a separate, unseen dataset to ensure the model’s ability to generalize effectively.

In the model development phase, a deep learning architecture—specifically the CNN-GRU (Convolutional Neural Network–Gated Recurrent Unit) model—is designed to effectively learn and represent complex patterns within the dataset. The model is first trained on the prepared training data and then evaluated to ensure optimal performance. Once the performance of the Cyber_Guard model meets the desired criteria, the trained model is saved for future inference. For prediction, the user provides test data, which is then processed by loading the pre-trained model to generate predictions. The results, indicating the identified attack type (such as DoS,

DDoS, or benign), are analyzed to gain meaningful insights into network activity.

To ensure ease of use, a graphical user interface (GUI) is developed using Python’s Tkinter module. The GUI is designed to be intuitive and interactive, featuring two primary sections: an input panel and a results panel. The input panel enables users to upload preprocessed test data in CSV format and trigger the prediction process with a single click. Once initiated, the system loads the trained Cyber_Guard model, processes the input, and displays the corresponding predicted attack type in the results panel, facilitating straightforward and efficient analysis.

STEPS are:

1) Data Pre-processing
- Loading dataset
- Handling missing values, if any
- remove unwanted columns

2) Feature Selection
- Initialize Feature Selection (ML method)
- Perform Feature Selection using Boruta

3) Data Balancing using SMOTE
- Initialize SMOTE algorithm
- Apply data balancing

4) Train-Test splitting in the ratio 80:20

5) Building Deep learning model CNN-GRU
- Defining various layers of CNN-GRU Model (Convolution Neural Network-Gated Recurrent Unit
- Create CNN-GRU architecture

6) Training the model
- Training the models with training dataset
- Performance Evaluation
- Save trained model Cyber_Guard

7) Predicting the output
- Input test data
- Loading the saved trained Cyber_Guard model
- Prediction using the Cyber_Guard model
- View result

The CNN-GRU model was selected for its strong ability to capture both spatial and temporal dependencies within network traffic data—an essential factor for accurate IoT attack detection. While CNNs are proficient in extracting spatial features from structured inputs, they lack the ability to model sequential dependencies over time. Conversely, models like RNNs and LSTMs handle temporal patterns effectively but are less efficient at learning spatial representations and often come with increased computational complexity. GRUs, as a streamlined variant of LSTMs, offer a comparable level of performance while using fewer parameters and enabling faster train- ing. By combining convolutional layers for spatial feature extraction with GRU units for sequence modeling, the CNN-GRU hybrid architecture capitalizes on the strengths of both approaches. This design proves particularly effective for processing the multi-dimensional and time-sensitive nature of IoT traffic data. Moreover, unlike deeper ResNet-based models that rely heavily on hierarchical feature abstraction and demand substantial computational resources, CNN-GRU provides a more lightweight, interpretable, and deployment-friendly solution—making it highly suitable for real- world IoT environments with limited resources.

4. System Architecture

A system architecture is a conceptual framework that outlines the structure, behavior, and key elements of a system. It typically comprises integrated subsystems and components that function collectively to achieve the system’s objectives. The following architectural diagram illustrates the workflow of the proposed system, highlighting the connections involved in both the training and prediction phases.

This architecture represents a complete pipeline for processing data and building a hybrid CNN-GRU model aimed at detecting various attack types. The process starts with the raw dataset, which undergoes a series of data preprocessing steps to clean and prepare it for analysis. After preprocessing, feature selection is applied to extract the most relevant attributes essential for accurate prediction. To address class imbalances, data balancing techniques are used, ensuring all classes are fairly represented.

The refined dataset is then divided into training and testing subsets. The training data is used to develop a hybrid deep learning model that integrates Convolutional Neural Networks (CNN) for extracting spatial features and Gated Recurrent Units (GRU) for capturing temporal patterns. Once trained, this model is capable of identifying and classifying various types of attacks with high accuracy.

In the prediction phase, the trained model is utilized to detect the type of DoS or DDoS attack. The prediction results are displayed through a graphical user interface (GUI) built using Python's Tkinter library.

5. Evaluation and Results

In assessing the performance and results of our model Cyber_Guard, various metrics such as accuracy, precision, recall, and F1 score are employed. These metrics serve as indicators of the model's classification capabilities and are derived from a confusion matrix .For evaluating the effectiveness of our proposed model, we rely on the following key metrics:

A. Accuracy: This metric measures the fraction of correctly classified applications relative to the total number of applications.

$$\text{Accuracy } = \frac{\text{TP+TN}}{\text{TP+TN+FP+FN}} \tag{1}$$

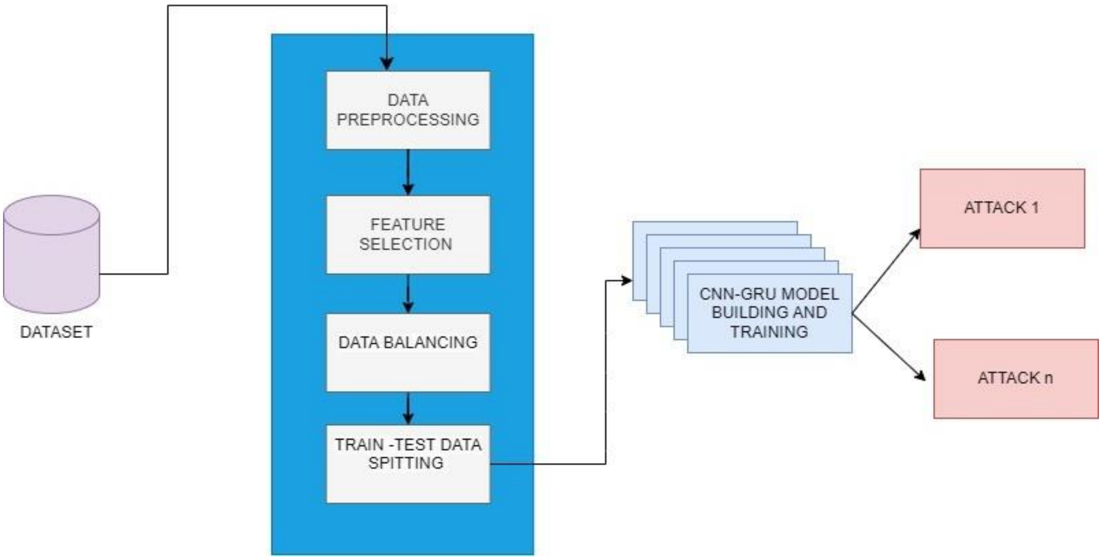


Fig. 2. System Architecture

B. Precision: Precision signifies the proportion of correctly predicted positive classifications to the total number of applications that are correctly identified as positive.

$$\text{Precision } = \frac{\text{TP}}{\text{TP+FP}} \tag{2}$$

C. Recall: also known as sensitivity or true positive rate, represents the fraction of correctly predicted positive classifications relative to the total number of correctly or incorrectly classified positive applications

$$\text{Recall} = \frac{\text{TP}}{\text{TP+FN}} \tag{3}$$

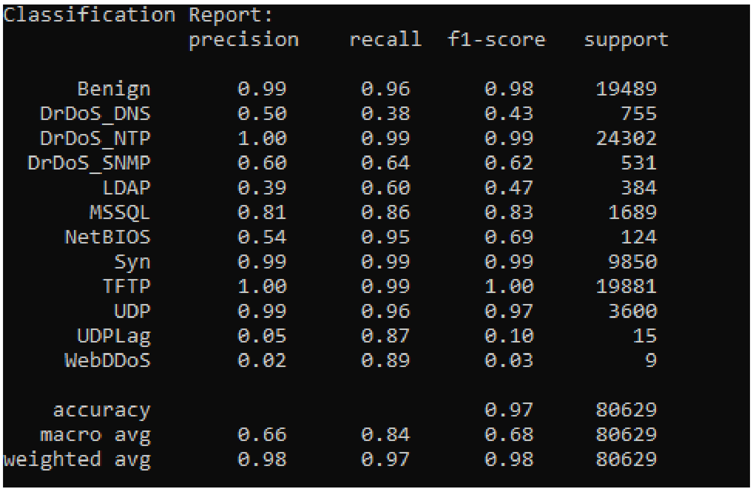
D. F-score: also referred to as the F1 score, is the harmonic mean of precision and recall.

$$\text{F1 Score} = 2 \times \frac{\text{Precision} \times \text{Recall}}{\text{Precision} + \text{Recall}} \tag{4}$$

These metrics collectively provide a robust framework for evaluating the performance and efficacy of our model Cyber_Guard in detecting DoS and DDoS attack accurately.

The performance matrix table, entitled "Performance Matrix" offers a detailed assessment of a classification model's performance across various classes. Each row within the table corresponds to different attacks while the columns delineate key evaluation metrics: precision, recall, F1-score, and support. Following is some of the conclusions that can be reached from the performance matrix table.

1) The Cyber Guard model achieved an accuracy of 97.06%, reflecting its effectiveness in correctly classifying the majority of samples
2) Precision value of 0.99 for the Syn class implies that 99% of instances predicted as Syn were correctly classified.
3) A recall value of 0.99 for the Syn class indicates that 99% of the actual Syn instances were captured by the model.
4) An F1-score of 0.99 for the Syn class indicates a harmonized balance between precision and recall, suggesting an optimal performance equilibrium.
5) A support value of 9850 for the Syn class signifies the presence of 9850 instances of the **Syn** category within the dataset.
6) For DrDoS_NTP: Precision value is 1.00 which means that out of all instances predicted as DrDoS_NTP, 100% were correctly identified as DrDoS_NTP
7) The Cyber_Guard model performs exceptionally well for some classes, like DrDoS_NTP and **Syn** with precision, recall, and F1-scores close to 1.
8) For less frequent classes like LDAP the performance metrics are lower, indicating potential room for improvement.
9) Cyber_Guard Model has perfect precision and recall for DrDoS_NTP but performs poorly on DrDoS_SNMP and UDP attacks.

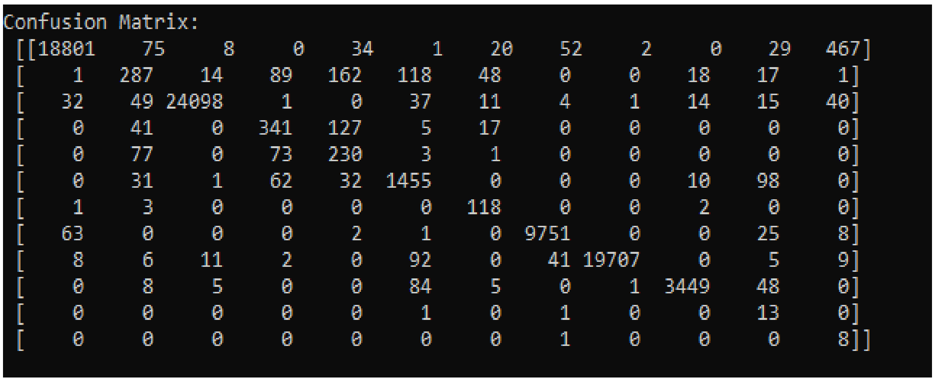


```
Classification Report:
              precision    recall  f1-score   support

      Benign       0.99      0.96      0.98     19489
  DrDoS_DNS       0.50      0.38      0.43       755
  DrDoS_NTP       1.00      0.99      0.99     24302
 DrDoS_SNMP       0.60      0.64      0.62       531
       LDAP       0.39      0.60      0.47       384
      MSSQL       0.81      0.86      0.83      1689
    NetBIOS       0.54      0.95      0.69       124
        Syn       0.99      0.99      0.99      9850
       TFTP       1.00      0.99      1.00     19881
        UDP       0.99      0.96      0.97      3600
     UDPLag       0.05      0.87      0.10        15
    WebDDoS       0.02      0.89      0.03         9

    accuracy                           0.97     80629
   macro avg       0.66      0.84      0.68     80629
weighted avg       0.98      0.97      0.98     80629
```

Fig. 3. Performance Metrices



```
Confusion Matrix:
 [[18801    75     8     0    34     1    20    52     2     0    29   467]
 [     1   287    14    89   162   118    48     0     0    18    17     1]
 [    32    49 24098     1     0    37    11     4     1    14    15    40]
 [     0    41     0   341   127     5    17     0     0     0     0     0]
 [     0    77     0    73   230     3     1     0     0     0     0     0]
 [     0    31     1    62    32  1455     0     0     0    10    98     0]
 [     1     3     0     0     0     0   118     0     0     2     0     0]
 [    63     0     0     0     2     1     0  9751     0     0    25     8]
 [     8     6    11     2     0    92     0    41 19707     0     5     9]
 [     0     8     5     0     0    84     5     0     1  3449    48     0]
 [     0     0     0     0     0     1     0     1     0     0    13     0]
 [     0     0     0     0     0     0     0     1     0     0     0     8]]
```

Fig. 4. Confusion Matrix

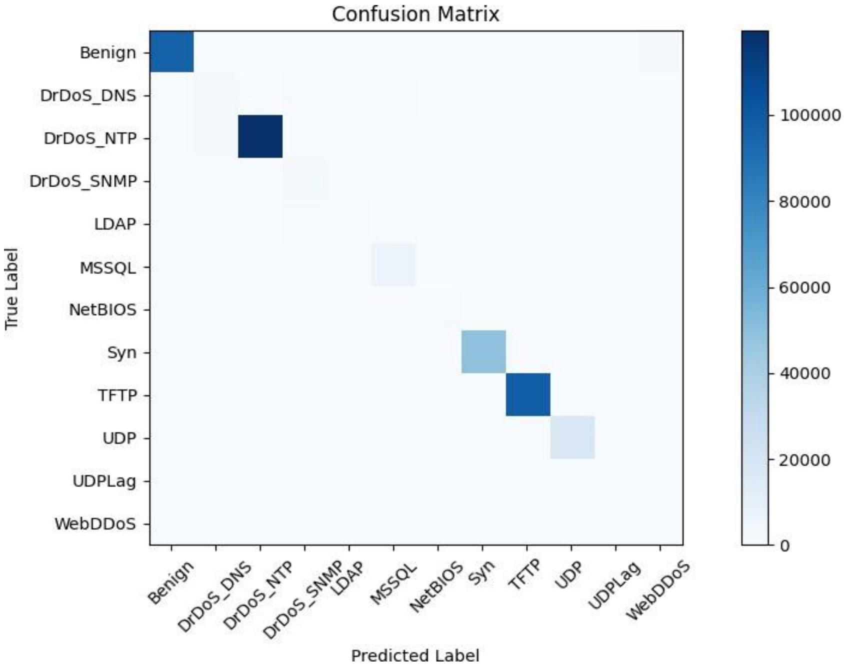


Fig. 5. Confusion Matrix

A table used to assess a classification algorithm's performance is called a confusion matrix. It shows the actual versus the predicted classifications, which allows us to see not only the accuracy of the model but also the types of errors it makes.The provided confusion matrix evaluates the performance of a multi-class classification model with 11 classes (0 to 10).

1) Diagonal Dominance: Most of the values are concentrated on the diagonal, indicating that most predictions are correct.
2) Class 0: Very high number of correct predictions (18801) with relatively few misclassifications.
3) Class 2: High correct predictions (24098) with minor misclassifications.
4) Class 7 and 8: High correct predictions (9751 and 19707 respectively).
5) Some classes have notable off-diagonal values indicating specific misclassification issues.
6) Class 10 has the fewest correct predictions (8), indicating potential difficulty in predicting this class accurately.
7) Class 1 has significant misclassifications with classes 4 and 5.
8) Class 3 has notable misclassifications with class 4.
9) Overall, the model performs well on several classes but struggles with certain classes, indicating areas for improvement in the classification process.
10) The first row and first column element (18801) means that there are 18801 instances that were correctly classified as class 0.
11) The first row and second column element (75) means that 75 instances that belong to class 0 were incorrectly classified as class 1.
12) The last row and last column element (8) mean that 8 instances that belong to class 10 were correctly classified as class 10.
13) The last row and first column element (0) mean that there were no instances that belong to class 10 that were incorrectly classified as class 0.

A comprehensive evaluation of the Cyber_Guard model's performance in detecting DoS and DDoS attacks is illustrated through the ROC curve analysis. Each colored curve in the plot corresponds to a specific attack class or benign traffic, with the True Positive Rate (TPR) on the Y-axis and the False Positive Rate (FPR) on the X-axis. The diagonal line represents the baseline performance of a random classifier. Most of the ROC curves cluster near the upper-left corner of the plot, indicating high sensitivity and low false positive rates. The Area Under the Curve (AUC) scores, which quantify the model's discriminatory power, are close to 1.0 for most classes. In particular, classes such as Benign, DrDoS_NTP, DrDoS_SNMP, LDAP, MSSQL, NetBIOS, Syn, TFTP, UDP, and UDPLag achieved perfect AUC scores of 1.0, reflecting flawless classification. Even for more challenging classes like DrDoS_DNS and WebDDoS, the model attained AUC values of 0.98, confirming its robustness and high reliability in differentiating between various forms of malicious and legitimate network.

The figure below illustrates the training and validation accuracy of the CNN-GRU model across five epochs. The blue curve denotes the training accuracy, while the red curve represents the validation accuracy. Both curves show a steady increase, indicating that the model is effectively learning over time. Initially, the training accuracy is around 0.86 and rises to approximately 0.94 by the fifth epoch. Similarly, the validation accuracy improves from about 0.78 to roughly 0.88. This upward trend in both training and validation accuracy suggests that the model is successfully capturing key patterns in the data and is not overfitting. The relatively small gap between the two accuracy curves highlights the model's ability to generalize well to unseen data. Overall, the graph reflects the strength and reliability of the Cyber_Guard model in accurately detecting DoS and DDoS attacks.

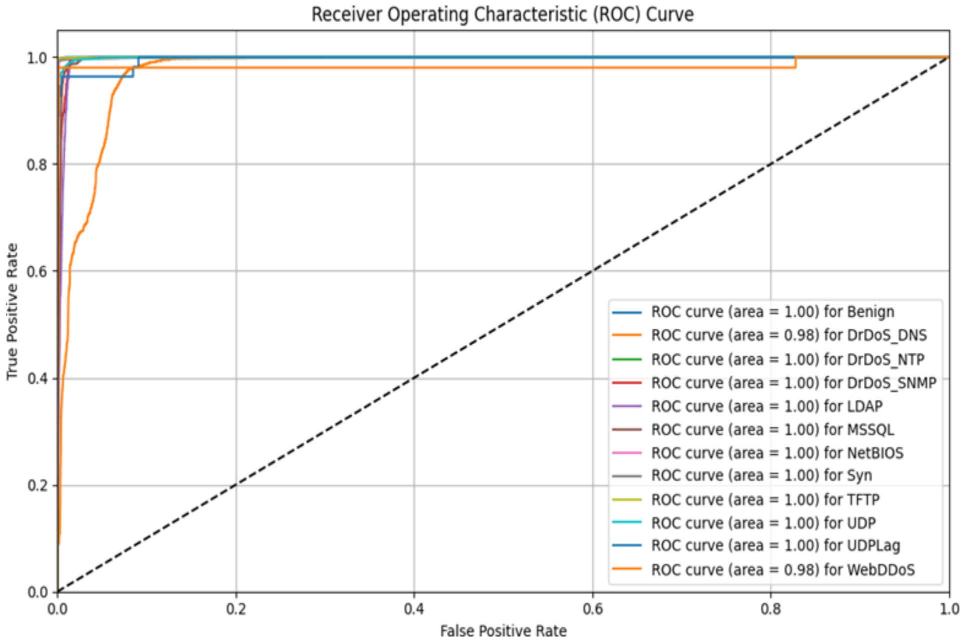


Fig. 6. ROC Curve

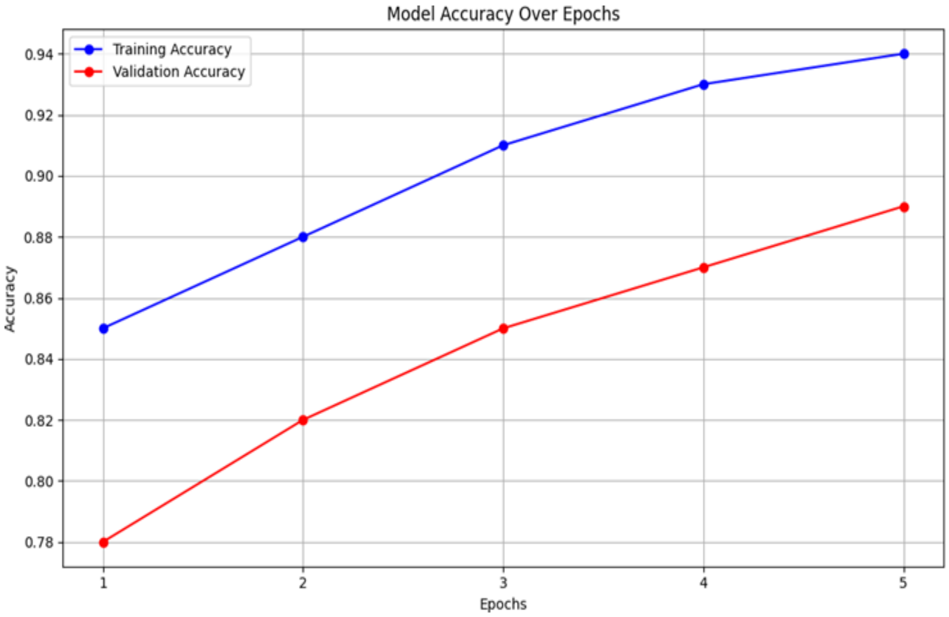


Fig. 7. Accuracy Curve

To optimize the CNN-GRU model's performance, critical hyperparameters were carefully adjusted through systematic trial and error. The tuning process explored different values for the learning rate, batch size, number of epochs, dropout rate, convolutional filter count, kernel size, and GRU unit quantity. After evaluating multiple configurations, the combination of a batch size of 64 and a learning rate of 0.001, paired with the Adam optimizer, was found to deliver the best results. This setup offered an ideal trade-off between training efficiency and model accuracy, ensuring stable convergence and improved detection performance.

Training was intentionally limited to five epochs to encourage generalization and minimize the risk of overfitting, a strategy that was supported by consistent validation performance. To further guard against overfitting, a dropout rate of 0.3 was applied to the fully connected layers. The final model architecture featured two convolutional layers with ReLU activation, followed by a GRU layer containing 64 units. These design choices were informed by expert knowledge and iterative empirical testing, with the objective of achieving an optimal balance between model accuracy, training efficiency, and computational cost.

Although the CICDDoS2019 dataset serves as a valuable and well-labeled source of IoT- specific network traffic, it is fundamentally a simulated dataset and may not encompass the full spectrum of real-world network variability, evolving attack techniques, or heterogeneous device behaviors. This limitation poses a challenge to the generalizability of the trained model, especially when confronted with previously unseen attack types or irregular traffic patterns in live operational environments. To improve the model's robustness and practical effectiveness, future research should prioritize testing on a broader range of datasets and consider deployment in real-time network settings, enabling a more accurate assessment of its adaptability and resilience under realistic conditions.

6. Comparative Study

The table compares the CNN-GRU model with the ResNet model [11] for detecting DoS and DDoS attacks in IoT environments. The CNN-GRU model integrates Convolutional Neural Networks (CNN) and Gated Recurrent Units (GRU) to effectively capture spatial and temporal features, achieving a high accuracy of 97%. This model benefits from the Boruta algorithm for feature selection, enhancing its ability to focus on relevant attributes, though it requires moderate effort to implement due to the complexity of combining CNN and GRU. On the other hand, Hussain's ResNet model, adapted from image recognition tasks, achieves an accuracy of 87%, slightly lower than the proposed model. The ResNet approach does not specify a feature selection method, potentially relying on its deep learning capabilities for direct feature extraction from the CICDDoS2019 dataset. Despite its effectiveness, the ResNet model's complexity results in lower ease of implementation compared to the proposed model. Both models use the CICDDoS2019 dataset, providing a consistent basis for performance evaluation. Overall, while the CNN-GRU offers superior accuracy and a structured approach to feature selection, the ResNet model shows the challenges of deploying deep learning architectures in IoT attack detection scenarios.

Table 1. Comparison Table

Feature	Proposed Model	Faisal Hussain [2]
Architecture Used	CNN - GRU	ResNet
Accuracy	97%	87%
Feature Selection	Boruta algorithm	Not Specified
Ease of Implementation	Moderate	Low
Dataset	CICDDoS2019	CICDDoS2019

Although both CNN-GRU and ResNet were assessed using the CICDDoS2019 dataset, the CNN-GRU model consistently outperformed ResNet in terms of accuracy and reliability across diverse attack categories. ResNet, originally built for image classification, benefits from residual connections that allow for deeper networks, yet it lacks the ability to model temporal dynam- ics—a critical requirement when dealing with sequential network traffic data [12]. In contrast, CNN- GRU combines the strengths of convolutional layers for spatial feature extraction with GRU layers for capturing temporal dependencies, making it more adept at identifying subtle attack patterns over time. The evaluation revealed that CNN-GRU delivered higher precision and recall across most classes, whereas ResNet struggled, especially with minority classes, often confusing attacks with similar traffic patterns like DrDoS DNS and LDAP. This shortcoming was primarily due to ResNet's lack of temporal learning capabilities. Moreover, CNN-GRU's moderate model complexity made it more efficient to train and fine-tune, while ResNet incurred significantly higher computational costs without yielding proportional improvements in detection performance.

7. Conclusion

According to Liang (2016), a denial-of-service attack is one that can be used to disrupt a network's connection and prevent authorised users from accessing it. Too many service requests from different systems are sent to the server during a DDoS attack [13]. The server gets so busy that it is unable to answer any of the service requests after receiving so many of them [8].In conclusion, this study explores the multifaceted challenges of IoT security, focusing on the detection of DoS and DDoS attacks using deep learning models—specifically CNN and GRU—augmented by the SMOTE algorithm to address class imbalance. A comprehensive review of existing literature reveals a heightened awareness of security vulnerabilities in IoT networks and underscores the necessity for advanced and adaptive intrusion detection systems. The combined strengths of CNN and GRU in capturing both spatial and temporal features, along with the data-balancing capability of SMOTE, highlight the potential for creating robust and reliable defense frameworks. Despite meaningful progress in this area, persistent challenges and evolving attack vectors call for ongoing research and innovation to further strengthen IoT security in real-world deployments.

8. Future Scope

DoS and DDoS attacks continue to pose critical risks in the IoT landscape, and existing detection mechanisms often struggle due to outdated datasets and the inefficiencies of models like CNNs when dealing with low-dimensional, non-visual data. To enhance the Cyber Guard system, future developments aim to achieve real-time threat detection by integrating live traffic capture with the trained CNN-GRU model, enabling continuous monitoring and rapid identification of attacks. Implementing this system on edge devices or IoT gateways, along with robust anomaly detection and alerting capabilities, can facilitate automated defensive responses. Nonetheless, key challenges such as improving inference speed, enabling real-time preprocessing, analyzing encrypted traffic, and seamless integration with intrusion prevention systems (IPS) must be carefully addressed.

References

[1] Hittu Garg and Mayank Dave. Securing iot devices and securelyconnecting the dots using rest api and middleware. In *2019 4th International Conference on Internet of Things: Smart Innovation and Usages (IoT-SIU)*, pages 1–6. IEEE, 2019.

[2] Yao Jiang, Kang Feng Zheng, Yi Xian Yang, Shou Shan Luo, and Jian Peng Zhao. Evaluation model for dos attack effect in softswitch network. In *2010 International Conference on Communications and Intelligence Information Security*, pages 88–91. IEEE, 2010.

[3] Faisal Hussain, Syed Ghazanfar Abbas, Muhammad Husnain, Ubaid U Fayyaz, Farrukh Shahzad, and Ghalib A Shah. Iot dos and ddos attack detection using resnet. In *2020 IEEE 23rd International Multitopic Conference (INMIC)*, pages 1–6. IEEE, 2020.

[4] Hanifatul Insan, Sri Suryani Prasetiyowati, and Yuliant Sibaroni. Smote-lof and borderline-smote performance to overcome imbalanced data and outliers on classification. In *2023 3rd International Conference on Intelligent Cybernetics Technology & Applications (ICICyTA)*, pages 136–141. IEEE, 2023.

[5] S Santhosh Kumar and K Kulothungan. An anomaly behavior based detection and prevention of dos attack in iot environment. In *2017 Ninth International Conference on Advanced Computing (ICoAC)*, pages 287–292. IEEE, 2017.

[6] Lulu Liang, Kai Zheng, Qiankun Sheng, and Xin Huang. A denial of service attack method for an iot system. In *2016 8th international conference on Information Technology in Medicine and Education (ITME)*, pages 360–364. IEEE, 2016.

[7] Suman Nandi, Santanu Phadikar, and Koushik Majumder. Detection of ddos attack and classification using a hybrid approach. In *2020 Third ISEA Conference on Security and Privacy (ISEA-ISAP)*, pages 41–47. IEEE, 2020.

[8] Yan Naung Soe, Paulus Insap Santosa, and Rudy Hartanto. Ddos attack detection based on simple ann with smote for iot environment. In *2019 fourth international conference on informatics and computing (ICIC)*, pages 1–5. IEEE, 2019.

[9] Krushang Sonar and Hardik Upadhyay. A survey: Ddos attack on internet of things. *International Journal of Engineering Research and Development*, 10(11):58–63, 2014.

[10] Shreekh Wankhede and Deepak Kshirsagar. Dos attack detection using machine learning and neural network. In *2018 Fourth International Conference on Computing Communication Control and Automation (ICCUBEA)*, pages 1–5. IEEE, 2018.

[11] Shruti Kajwadkar and Vinod Kumar Jain. A novel algorithm for dos and ddos attack detection in internet of things. In *2018 Conference on Information and Communication Technology (CICT)*, pages 1–4. IEEE, 2018.

[12] S Yuvalatha, S Nithyapriya, R Tamizh Kuzhali, and S Savitha. Boruta feature selection for prediction of coronary artery disease. In *2023 International Conference on Research Methodologies in Knowledge Management, Artificial Intelligence and Telecommunication Engineering (RMKMATE)*, pages 1–5. IEEE, 2023.

[13] I Varalakshmi, M Thenmozhi, and R Sasi. Detection of distributed denial of service attack in an internet of things environment-a review. In *2021 international conference on system, computation, automation and networking (ICSCAN)*, pages 1–6. IEEE, 2021.

Authors' Profiles

Jinsu Anna John currently working as an Assistant Professor in the Department of Computer Science and Engineering (Cyber Security) at Vimal Jyothi Engineering college Kannur.

Raj Kumar T. is currently serving as an Assistant Professor in the Department of Computer Science and Engineering (Cyber Security) at College of Engineering Kalloopara, Kerala, India.

Shilpa Harrison is currently serving as an Assistant Professor in the Department of Computer Science and Engineering (Cyber Security) at College of Engineering Kalloopara.

How to cite this paper: Jinsu Anna John, Raj Kumar T., Shilpa Harrison, "Cyber Guard: Detecting DoS and DDoS Attack", International Journal of Wireless and Microwave Technologies(IJWMT), Vol.15, No.5, pp. 57-67, 2025. DOI:10.5815/ijwmt.2025.05.05