

# **A CNN-GRU Based Optimized and Explainable Framework for TCP-UDP DDoS Attack Detection and Deployment on Edge Devices**

Joshi Shraddha<sup>1</sup> and Dr.Vikram Agrawal<sup>2</sup>

1. Research Scholar, Gujarat Technological University, Ahmedabad, Gujarat, India.
2. Assistant Professor at BBIT, Gujarat Technological University, Vallabh Vidyanagar, Anand, Gujarat, India.

## **Abstract**

Distributed Denial of Service (DDoS) attacks have been targeting IoT and IIoT systems more and more, which calls for the development of accurate, efficient, and interpretable detection methods. With particle swarm optimization (PSO) for hyperparameter selection and SHAP-based Explainable AI (XAI) for explainability, this paper proposes a hybrid CNN-GRU model employing CNNs for spatial feature extraction and GRUs for temporal pattern learning. Applied to the Edge-IIoT dataset, UDP model demonstrated ideal recall (100%) with less false positives, whereas the TCP model was enhanced from the baseline of 65.47% accuracy and 59.58% precision to 99.82% accuracy and 99.70% precision after PSO optimization. As per the SHAP study, predictions are affected by important factors such as TCP flags and DNS retransmissions. Finally, deployment on Raspberry Pi 5 ensured real-time feasibility by delivering performance equivalent to server execution. For DDoS detection in IoT/IIoT networks, the proposed framework guarantees a trustworthy, explainable, and edge-deployable solution.

**Index Terms** IoT Security, IIoT, DDoS Detection, Deep Learning, CNN-GRU, Hyperparameter Optimization, Explainable AI.

## **I. INTRODUCTION**

The Internet Of Things (IoT) and Industrial IoT (IIoT) have revolutionized industries, health care, smart spaces through networking devices to communicate easily between each other, applications and services etc. [1]. As a result of their widespread adoption, IoT networks tend to be more prone to cyber-attacks involving Distributed Denial of Service (DDoS) attacks in particular [2]. Overloads of heavy network flows due to DDoS attacks overflow systems causing large down-times, unavailability and losses. The increasing complexity and quantity of orthodox detection methods bring about tough challenges for their incapability to reach the real-time accuracy, scalability, flexibility for IoT dynamic environments [3]. A typical Internet of things/Industrial internet of things architecture pyramid is shown in Fig. 1.

Recent advances in Machine Learning (ML) and Deep Learning (DL) have opened a promising perspective to enhance the detection of DDoS attacks by learning traffic patterns automatically from raw data [4]. Classic ML methods, such as Support Vector Machines (SVM), Random Forests or Naïve Bayes classifiers are not so relevant when used to solve the problem of high-dimensional and heterogeneous IoT traffic because they rely on explicit features [5]. On the other hand, the DL models, specifically the Convolutional Neural Networks (CNNs) and Recurrent Neural Networks (RNNs) [6], have been found to be more capable of recognizing complex patterns in the network traffic. High computational expenses, poor hyperparameter tuning, and poor interpretability are some of the issues that complicate the use of deep learning models in IoT environments of limited resources.

Our study proposed a hybrid CNN–GRU framework for efficient TCP and UDP-based DDoS detection in order to address these problems. While the GRU module records temporal relations to identify unusual traffic patterns over time, the CNN module learns local geographical correlations among packet features. In order to significantly increase detection efficiency and decrease false positives, Particle Swarm Optimisation (PSO) is introduced to provide automatic hyperparameter tuning. Explainable AI (XAI) based on SHAP values also goes beyond the “black-box” nature of deep models by offering security experts useful insights and feature-level interpretability.

The implementation of a Raspberry Pi edge device to verify the proposed paradigm in practice is another significant contribution of this work. In distributed IoT networks, edge implementation guarantees real-time attack detection closer to data sources, lowers latency, and permits scalability. Experiments using the publicly accessible Edge-IIoT dataset show that the optimised CNN–GRU model outperforms its baseline configuration and achieves nearly perfect accuracy and recall rates following PSO optimisation.

The primary contributions of this study can be stated as follows:

- Design of a CNN–GRU hybrid architecture for IoT/IIoT networks’ TCP-UDP DDoS detection
- Hyperparameter optimization, enabled with PSO, was integrated into this model, resulting in higher model accuracy and lowering false positives. SHAP is used to generate feature-level insights and improve the reliability of model predictions.

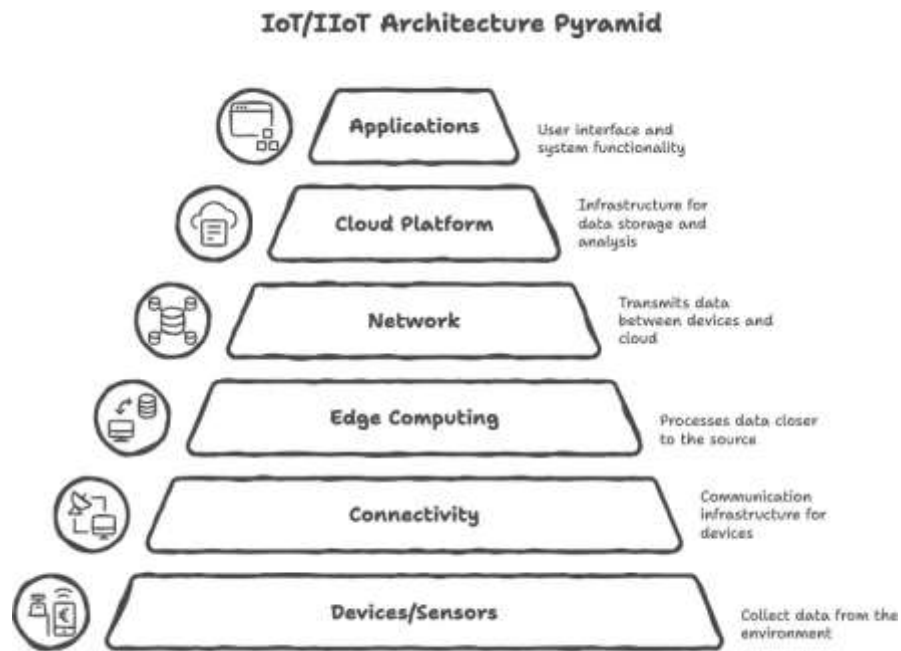


**Fig. 1: IoT/IIoT architecture pyramid**

- Raspberry Pi edge deployment, proving viability in resource-constrained real-world IoT scenarios.
- A thorough analysis utilizing the Edge-IIoT dataset revealed that it performed better than the baseline model.

The structure of the paper is as follows: The related work on DDoS detection techniques is reviewed in Section 2. In Section 3, the dataset and preprocessing methods are presented. Details of the suggested CNN-GRU model and optimization techniques are provided in Section 4. The experimental findings and performance assessment are shown in Section 5. The setup of the edge deployment and the results are covered in Section 6. Section 7 brings the work to a close and suggests areas for further research.

## **II. RELATED WORK**

The present body of research can be broadly classified as explainable AI methods, optimization-based models, deep learning-based solutions, edge-based security systems, and classical ML methods.

To detect Distributed Denial of Service (DDoS) attacks in 5th Generation (5G) telecommunication networks, the development and validation of advanced Machine Learning (ML) and Deep Learning (DL) algorithms is discussed in [7]. Effective detection systems are needed since 5G technologies expose networks to cyberattacks threatening service integrity. The main goal of this study is to develop and test ML and DL algorithms that can successfully identify DDoS attacks in 5G telecommunication networks. Through the process of processing data in real-time, these algorithmic approaches will enhance robustness of network security protocols and improve cyber threat resistance. A stable simulated environment was built using free 5GC and UERANSIM [7] that models the complex interdependencies of 5G networks. This allows for easier comparison of ML and DL models under normal as well as attack conditions. The models proposed or evaluated include XGBoost, Random Forest, Support Vector Machine (SVM), Bidirectional Encoder Representations from Transformer (BERT), Bidirectional Long Short-Term Memory (BiLSTM), Multilayer Perceptron (MLP), and a Custom Convolutional Neural Network (CNN) [7]. Better performance of Random Forest and XGBoost model makes it suited for dynamic 5G environment. Thus, this comparative study intends to identify the most effective real-time DDoS-detection methods that ensure the highest accuracy and low complexity while also effectively utilizing resources in the network. According to the results, the customised ML, DL and ensemble models can be very efficient for detecting DDoS attacks, with superior real-time threat detection performance and accuracy [7]. It shows how these models can be turned into useful models within actual telecom networks of the present day.

While Software-Defined Networking (SDN) can give an added layer of control and facilitate programming, its centralised nature makes it susceptible to Distributed Denial-of-Service (DDoS) attacks [8]. To overcome this limitation, [8] proposes a detection approach using Multilayer Perceptron (MLP) and Convolutional Neural Network (CNN) algorithms. This is reinforced by Bayesian optimisation for feature selection via SHAP and hyperparameter tuning. Bayesian optimisation contributes to improved model performance and reduced false positive rates, SHAP enhances interpretability by finding the most important

features in DDoS detection. Later, it verified accuracy via CICDDoS-2019 and InSDN datasets by having a true positive rate of 99.95% and 99.98%, respectively.

On one hand, Software-Defined Networking (SDN) has a centralised control setup that exposes it to Distributed Denial of Service (DDoS) attacks. To identify such attacks, [9] proposes a CNN-GRU-based DL architecture for identifying spatial patterns and temporal correlations in network data. A variety of datasets, including attack and normal traffic, were reviewed to evaluate the model against standard and popular machine learning algorithms. According to benchmark measures, the CNN-GRU model systematically outperforms typical approaches in terms of performance [9]. It presents good accuracy, stability, and overall performance, which makes it very suitable to shield SDN against the dynamic DDoS attacks.

Due to the rapid growth of IoT edge devices, the security landscape has developed, where Distributed Denial of Service (DDoS) attacks have grown more common and complex. For example, legacy destination-based defenses may sometimes find high false positives, as the malicious traffic may appear legitimate, and source-only defenses fall short because they have only partial awareness of ongoing attacks [10]. SmartDefense, this study proposes a distributed DDoS detection and mitigation system based on edge computing concepts to help mitigate the issues of the above mentioned limitations [10]. SmartDefense allows detection and response at or close to the source of the attack, allowing for early intervention prior to the attack traffic requiring substantial resources. By stopping malicious traffic at home edge networks, SmartDefense eliminates unnecessary bandwidth consumption across ISP networks, providing a more scalable and effective way of fighting today's DDoS attacks [10].

The Internet of Things (IoT) is transforming precision healthcare through cost reductions and enhanced patient care, but in 5G networks, healthcare-IoT (H-IoT) devices are yet vulnerable to Distributed Denial of Service (DDoS) attacks [11]. A detection and mitigation approach based on a temporal convolutional network (TCN) with dynamic thresholding and monitoring frequency is demonstrated in [11]. At 5-second intervals, the number of messages transmitted by each H-IoT node is tracked, and adaptive thresholds are determined using averages of hazardous behavior across nodes. Datasets produced by MQTT and UDP traffic in simulated 5G H-IoT scenarios (Cooja and ns-3) were used to test the framework. Outperforming the BiLSTM and CNN baselines, the results provide 99.98% accuracy on MQTT and 99.99% accuracy with 80% mitigation on UDP [11]. According to the results, TCN is a reliable, flexible, and efficient security solution for H-IoT systems.

Managing and protecting IoT devices over dispersed edge networks has become a crucial concern due to their explosive expansion in smart homes, cities, and enterprises. The necessity for efficient monitoring and defense is highlighted by recent hacks that take advantage of flaws in IoT devices [12]. In order to automatically gather inbound, outgoing, and internal traffic from IoT devices, this article presents an IoT traffic measurement framework that is installed on programmable, intelligent edge routers [12]. To describe device actions, the framework creates multidimensional behavioral profiles that capture temporal, geographic, entropy, and cloud service trends. [12] presents n important insights into protecting vulnerable and resource-constrained IoT devices in edge environments, as it is the first to our knowledge to systematically evaluate these multidimensional fingerprints for IoT device classification, anomaly detection, and security monitoring.

Despite its rapid growth, the heterogeneous Internet of things (HetIoT) is confronted with increasing security threats, including Distributed Denial of Service (DDoS) assaults [13]. In order to improve security and performance, an Edge-HetIoT Intrusion Detection System (IDS) to identify and stop harmful traffic close to the network edge is presented in [13]. A suggested deep learning hybrid model (CNN+LSTM) and five machine learning classifiers (ID3, Naïve Bayes, Random Forest, Logistic Regression, and AdaBoost) are among the six learning approaches that are analyzed. Performance is compared between models using the CICDDoS2019 dataset for binary and multiclass (14 classes) classification. With overall weighted averages of 92% precision, 89% recall, and 90% F1-score, the hybrid CNN+LSTM model surpasses all ML baselines in identifying MSSQL, NetBIOS, TFTP, NTP, SYN, and Portmap attacks, obtaining 100% precision, recall, and F1-score [13].

In the current digital era, network security is still seriously threatened by Distributed Denial of Service (DDoS) attacks, especially in Software-Defined Networks (SDNs) [14]. [14] provides a detection system based on Deep Neural Networks (DNNs). DL is used by DNNs to examine network data and find intricate patterns connected to DDoS attacks. Tests conducted on real-world datasets such as InSDN, CICIDS2018, and Kaggle DDoS consistently demonstrated low loss rates and attained detection accuracy of 99.98%, 100%, and 99.99%, respectively [14]. The performance achieved by the proposed method compared to conventional detection methods is enhanced. On this basis the system stability, scalability and agility, are very effective in reducing modern DDoS attacks. As well as its excellent empirical performance [14], this work demonstrates the deployment challenges of SDN and provides practical suggestions for improving SDN security. The relevant works are shown in Table I.

### III. DATASET AND PREPROCESSING

#### A. Dataset Description

The Edge-IIoTset dataset [15], which has recently been established as an exhaustive and applicable benchmark for the IoT/IIoT cybersecurity, is used in this study. The dataset comprises packets of many protocols (TCP, UDP, DNS, MQTT and Modbus TCP), under benign case and malicious case like Distributed Denial of Service (DDoS) attacks.

TABLE I: Summary of Recent DDoS Detection Approaches in IoT

| Ref. | Method / Algorithm Highlights                                                                              | Application                                             | Results                                                                                     |
|------|------------------------------------------------------------------------------------------------------------|---------------------------------------------------------|---------------------------------------------------------------------------------------------|
| [7]  | BERT, Optimization<br>BiLST<br>M,<br>MLP, CNN, Random<br>Forest, SVM,<br>XGBoost, Ensemble<br>(RF+XGBoost) | DDoS detection in 5G net- works (free 5GC + UER- ANSIM) | Ensemble RF+XGBoost achieved highest accuracy<br>TPR: 99.95% (CICDDoS-2019), 99.98% (InSDN) |
| [8]  | CNN, MLP + SHAP<br>(fea-<br>ture selection) +<br>Bayesian                                                  | DDoS detection in SDN (CICDDoS-2019, InSDN)             |                                                                                             |

|      |                                                                                  |                                                                          |                                                                                                                              |                                                                                                                                                  |
|------|----------------------------------------------------------------------------------|--------------------------------------------------------------------------|------------------------------------------------------------------------------------------------------------------------------|--------------------------------------------------------------------------------------------------------------------------------------------------|
|      | Ensemble models improved detection but introduced higher                         | computational cost in resource-constrained 5G environments.              | SHAP improved interpretability; Bayesian                                                                                     | optimization reduced false positives and improved efficiency.                                                                                    |
| [9]  | CNN-GRU hybrid DL framework                                                      | DDoS detection in SDNOutperformed traditional ML and existing DL methods |                                                                                                                              | Effectively captured spatial and temporal dependencies in network traffic.                                                                       |
| [10] | SmartDefense (distributed edge computing framework)                              | DDoS mitigation in edge IoT environments                                 | Early mitigation near source reduced bandwidth consumption                                                                   | Addressed limitations of source-only and destination-only defense; scalable and efficient for ISP and                                            |
| [11] | Temporal Convolutional Networks (TCN) + monitoring frequency + dynamic threshold | DDoS detection and mitigation in 5G Healthcare- IoT (MQTT, UDP)          | 99.98% accuracy (MQTT); 99.99% accuracy, 80% mitigation (UDP) Generated multidimen- sional                                   | residential edge. Outperformed BiLSTM and CNN; adaptive thresholding eliminated false positives.                                                 |
| [12] | IoT traffic measurement framework (edge routers)                                 | IoT device traffic analysis and anomaly detection                        | behavioral profiles                                                                                                          | First to characterize IoT devices using tem- poral, spatial, entropy, and cloud service fin- gerprints for classification and monitoring. Hybrid |
| [13] | CNN+LSTM hybrid DL vs. ML (ID3, NB, RF, LR, AdaBoost)                            | (CICDDoS2019 dataset)                                                    | CNN+LSTM: 100% precision, recall, F1 for MSSQL, NetBIOS, TFTP, NTP, SYN, Portmap; overall: 92% precision, 89% recall, 90% F1 | CNN+LSTM outperformed all ML baselines in binary and 14-class classifica- tion.                                                                  |
| [14] | Deep Neural Network (DNN)                                                        | DDoS detection in SDN (InSDN, CICIDS2018, Kaggle DDoS)                   | Accuracies: 99.98% (InSDN), 100% (CICIDS2018), 99.99% (Kaggle)                                                               | Robust, scalable, adaptable; consistently outperformed conventional methods with low loss rates.                                                 |

The dataset, titled “Edge-IIoTset: A New Comprehensive Realistic Cybersecurity Dataset of IoT and IIoT Applications: Centralized and Federated Learning,” is made publicly accessible via IEEE DataPort<sup>1</sup>. Two preprocessed subsets, namely TCP and UDP of the dataset were used for this investigation. By including both typical traffic flows and DDoS attack traces, these subsets allow model performance to be assessed across two essential transport protocols.

### B. Data Distribution

Table II shows the general composition of the TCP and UDP subsets, which include both attack and normal cases. The total amount of samples may vary somewhat based on preprocessing, but both subsets are sufficiently balanced to enable reliable DDoS detection model training and evaluation.

TABLE II: Description of TCP and UDP Subsets from Edge-IIoTset

| Dataset    | Traffic Type | Description                                                                   |
|------------|--------------|-------------------------------------------------------------------------------|
| TCP Subset | Normal       | Regular TCP traffic generated by benign IoT/IIoT applications.                |
|            | Attack       | DDoS traffic targeting TCP connections (e.g., SYN floods, connection resets). |
| UDP Subset | Normal       | Standard UDP flows from benign IoT communication.                             |
|            | Attack       | Malicious UDP floods designed to exhaust bandwidth and disrupt services.      |

### C. Feature Engineering

There were 63 features in the original Edge-IIoT dataset that represented traffic at various network layers, including frame, IP, TCP/UDP, and application protocols. Zero-variance characteristics and non-informative identifiers (such IP addresses) were eliminated in order to cut down on noise and redundancy. The target label (Attack\_label) and 35 numerical features were kept after preprocessing.

### D. Preprocessing Steps

To prepare the raw network traffic features for deep learning, a number of preprocessing processes were carried out before the model was trained [16]. By taking these precautions, the CNN-GRU architecture was guaranteed to be able to capture the dataset’s temporal and spatial connections.

- 1) **Feature Normalization:** Features in the Edge-IIoT dataset have a variety of sizes, from huge numerical values (such sequence numbers and retransmission counts) to tiny binary indications (like flag variables). Large-magnitude features have the potential to dominate learning and skew the model if they are not normalized. Two normalizing techniques were used in various experimental configurations to address this:
  - **Decimal Scaling:** To bring values into a similar numerical range, they were scaled by dividing them by the relevant powers of 10. The relative distribution of features with

significant numerical variability (such as sequence numbers) was maintained using this strategy.

- **Min-Max Normalization:** To improve gradient stability and guarantee that every attribute contributed uniformly during training, each feature was rescaled to a fixed range [0,1].  
These normalization methods decreased the possibility of vanishing or exploding gradients in the recurrent layers and aided in speeding up convergence during optimization.
- 2) **Dataset Splitting:** The data was split in an 80:20 ratio to ensure an objective evaluation. The parameters of the CNN- GRU model were trained using the training data, and the evaluation of the performance was the only use of the testing data. The ratio of attack and normal instances in both sets was ensured to be balanced to prevent any issues of class imbalance that could affect the generalization of the model.
- 3) **Time-Series Input Formatting:** Network flow data has to be modeled as a sequence, not as a standalone entity, since there are strong temporal dependencies in DDoS traffic (e.g., SYN packet bursts or irregular timing between packets). The feature vectors were rearranged to form time-series input sequences for this, where each example was represented as a 3D tensor with dimensions: (batch size, time steps, features). In this manner, the GRU layers were able to effectively extract long-term temporal features (e.g., periodic retransmissions or sustained flooding activity), and the CNN layers were able to effectively model local spatial dependencies in the data, such as flag bits and port activity.
- 4) **Label Encoding:** The target variable (Attack\_label) was encoded with binary classes 0 (Normal) and 1 (Attack). The binary encoding of the target variable made the classification task easier while also being compatible with the sigmoid activation function in the final dense layer of the CNN-GRU model.

#### **IV. PROPOSED APPROACH METHODOLOGY**

In the IoT/IloT community, the proposed architecture aims to offer a trustworthy, efficient, and interpretable method for DDoS attack identification. The proposed method uses deep learning to identify complex patterns in network traffic automatically, instead of using predefined rules or manually designed features for conventional intrusion detection. In essence, three different components are merged by the architecture; the proposed method architecture is shown in Fig. 2.

- CNN-GRU hybrid deep learning architecture for learning temporal and spatial features.
- It uses PSO for hyperparameter tuning. Explainable AI (XAI) for improving the interpretability of the models and enhancing confidence in predictions.
- These elements work together to address the current three major issues of feature complexity, hyperparameter dependency, and lack of transparency in the IoT-based DDoS detection.

#### **Hybrid CNN-GRU for Spatio-Temporal Learning**

In fact, DDoS attacks may involve both temporal dependencies (such as repeated packet flooding over time) and spatial irregularities (such as irregular flag patterns and unexpected port usage). These properties cannot be captured by a single class of models.

- Local correlations and packet-level patterns are detected anywhere in the traffic stream by the CNN component, which serves as the spatial feature extractor.

- The GRU component handles the temporal aspect, learning traffic dynamics over time steps, and detecting long-term anomalies such as retransmissions and traffic spikes

The model achieves higher detection accuracy than standalone systems by integrating CNN and GRU into a single pipeline that captures both sequential behavioral patterns and instantaneous attack fingerprints.

### **PSO for Hyperparameter Optimization**

You can also see hyperparameters such as the number of filters in the CNN layers, GRU units, the learning rate, dropout probability, and batch size that affect deep learning performance. Manual tuning is ineffective. To counter this, the system employs Particle Swarm Optimization (PSO), a population-based metaheuristic modeled after the collective behavior of bird flocks.

- Each particle in the swarm can be considered a possible hyperparameter configuration.
- The swarm then iteratively explores the parameter space, in an optimization process directed by both local and global optimal solutions.
- At the end of the process, the desired collection of hyperparameters serves as the optimized sets that the model can use to perform well (accuracy, precision, recall).
- PSO automates the search process through which the CNN-GRU model is trained, ensuring that it follows the optimal configuration and does not require laborious manual tuning.

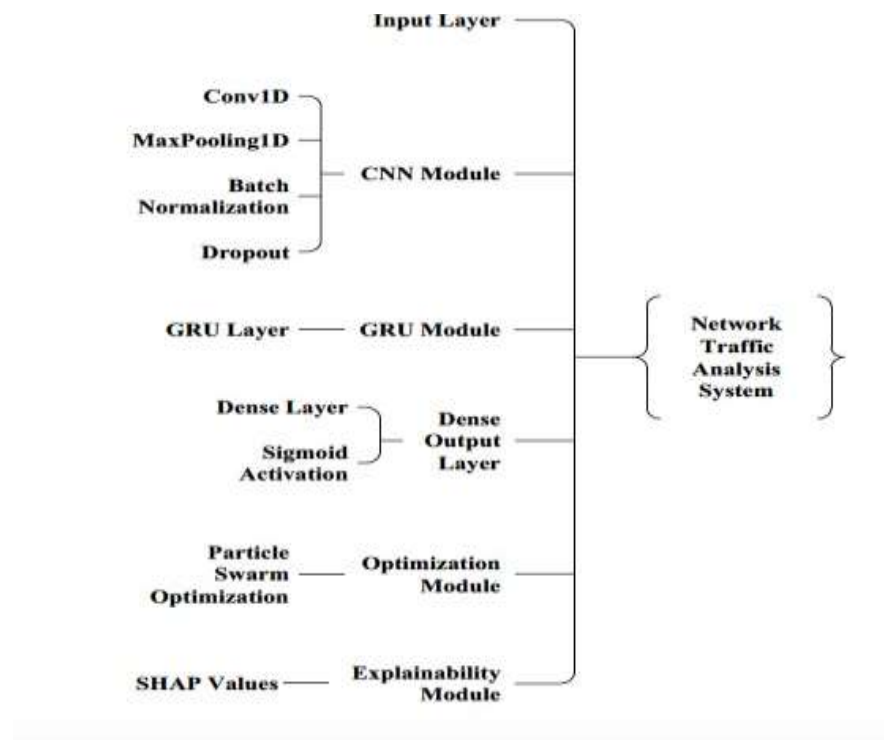


**Fig. 2: Proposed Approach Architecture**

### ***XAI for Interpretability***

Deep learning models can be labeled as "black boxes", even though they generate very good prediction results. Thus, managers must understand why a network flow is considered malicious to deploy effective countermeasures in the DDoS detection of security-critical applications. The proposed technique works by adopting the SHapley Additive exPlanations (SHAP) method for introducing Explainable AI (XAI) techniques to this issue.

- SHAP assigns a contribution score to each feature, which indicates whether the feature raised or lowered the probability of an attack prediction.
- Decision-making for individual examples is illustrated transparently in terms of individual samples by visualisations such as waterfall graphs.
- This model's emphasis on significant features in this way is also verified by global feature importance analysis, which identifies the prominent features throughout the dataset (such as TCP SYN flags or DNS retransmission counts).

In addition to increasing the model's credibility, its interpretability gives network defenders useful information.

### ***Framework Significance***

Combined, these three elements mitigate some of the deficiencies identified in the present literature that inform the proposed framework:

- CNN-GRU ensures comprehensive spatiotemporal feature extraction.
- PSO ensures that the best hyperparameters are chosen, enhancing performance and lowering false positives.
- By providing interpretability, XAI bridges the gap between actual application and predictive accuracy.

Furthermore, the framework is made to be sufficiently lightweight for deployment on edge devices (like the Raspberry Pi), allowing for distributed, scalable, and real-time DDoS detection in IoT/IIoT networks.

### ***A. CNN-GRU Hybrid Architecture***

Although the use of a single architecture often hinders the ability to fully grasp the whole range of patterns that exist within network data, deep learning techniques have shown considerable promise in the realm of intrusion detection. Although Gated Recurrent Units (GRUs) are masters at grasping temporal relationships [17], Convolutional Neural Networks (CNNs) are excellent at grasping spatial relationships between features [18]. The research applies a hybrid CNN & GRU model designed specifically for DDoS detection in the IoT/IIoT environment, in a way that allows it to leverage the complementary strengths of both models. The model applies a number of levels, each of which contributes to the learning process as a whole:

#### ***Input Layer:***

The input to the model is the preprocessed traffic data, which is arranged in the form of three-dimensional tensors with the following shape:

(batch size,time steps,features)

Where features represent the set of traffic features retained after preprocessing, batch

size represents the number of traffic sequences processed simultaneously, and time steps represent the flow of network traffic over time. The temporal and geographical dependencies required for accurate detection are preserved in this form.

**Convolutional Layer (Conv1D):**

As a method for extracting local spatial patterns in network traffic (e.g., odd port activity or flag combinations), the first hidden layer applies one-dimensional convolution to sequences of features.

- **Parameters:** The parameters are 21 filters, kernel size of 5, and activation function of ReLU.
- **Function:** This adds robustness to variations in traffic by learning patterns that are discriminative irrespective of their location in the sequence.

**MaxPooling1D Layer:**

Feature maps are dimensionally reduced by selecting the maximal activation inside each window using a max-pooling procedure after convolution

- **Parameters:** Pool size=2
- **Function:** Ensures that only dominant local features are preserved, improves computational efficiency, and lowers noise

**Batch Normalization Layer:**

For better convergence and stabilization during training, batch normalization re-centers and re-scales the activations of the preceding layer.

- **Function:** Prevents internal covariate shifts, permits better learning rates, and serves as a regularizer to lessen overfitting.

**Dropout Layer:**

Also, a dropout mechanism is introduced to avoid overfitting by randomly deactivating a portion of neurons during training.

- **Parameters:** Dropout Rate=0.46
- **Function:** Promotes the model's acquisition of broader patterns that are independent of particular neurons.

**GRU Layer:**

The Gated Recurrent Unit (GRU) captures temporal relationships across traffic data sequences to correctly depict attack patterns that evolve over time, including recurring retransmissions or persistent floods.

- **Parameters:** 14 units, return\_sequences = False
- **Function:** Unlike conventional LSTMs, it maintains computing efficiency while learning long-term sequential behaviors.

**Dense Output Layer:** A dense, fully linked layer with a sigmoid activation function for binary classification makes up the last layer.

- **Output:** Produces a probability score between 0 and 1, where a DDoS attack is indicated by a value closer to 1 and normal traffic by a value closer to 0. The summary for the parameters of proposed approach is presented in Table III.

**TABLE III: CNN-GRU Model Architecture and Parameters**

| Layer               | Description                                                                     | Key Parameters                                                                                | Purpose                                                      |
|---------------------|---------------------------------------------------------------------------------|-----------------------------------------------------------------------------------------------|--------------------------------------------------------------|
| Input Layer         | Preprocessed traffic sequences (3D tensor)                                      | Shape: (batch size, time steps, features)<br>Filters = 21, Kernel size = 5, Activation = ReLU | Preserves spatial and temporal information                   |
| Conv1D              | Convolutional layer for spatial patterns                                        |                                                                                               | Extracts local correlations (e.g., TCP flags, port activity) |
| MaxPooling1D = 2    | Down-sampling of feature maps<br>Reduces dimensionality, noise, and computation |                                                                                               | Pool size                                                    |
|                     | Normalizes intermediate activations<br>training, accelerates convergence        | -                                                                                             | Stabilizes                                                   |
| Dropout rate = 0.46 | Random deactivation of neurons<br>Prevents overfitting, improves generalization |                                                                                               | Dropout                                                      |
| GRU                 | Sequential modeling of temporal patterns                                        | Units = 14, return sequences = False<br>-                                                     | Learns attack patterns unfolding over time                   |
| Dense Output        | Final binary classification<br>Predicts probability of Normal (0) or Attack (1) | Units = 1, Activation = Sigmoid                                                               |                                                              |

**B. Hyperparameter Optimization with PSO**

Hyperparameters selection, specifically the number of convolutional filters, GRU units, learning rate, dropout rate, batch size, and so on, significantly impacts the performance of deep learning models. These factors significantly contribute to the model's capability to generalize on unseen traffic and identify patterns from data. However, hyperparameters are difficult to optimize manually because of the extremely complex nature of the search space, potentially resulting in low-quality solutions. To go beyond this limitation, this study uses Particle Swarm Optimization (PSO) as an automated hyperparameter optimization technique.

PSO is a population-based metaheuristic algorithm that takes inspiration from nature [19]. It is based on the behavior of schools of fish and flocks of birds [20]. In this case, every "particle" in the population represents a potential solution or a set of hyperparameters for the CNN-GRU model. To improve the performance of the model, every particle changes its position (hyperparameters) and velocity (speed of change) as it traverses through the search space.

**Optimization Process:**

- 1) Initialization: Random hyperparameter values (such as learning rate, number of filters, and dropout rate) are used to initialize each of the created particles.
- 2) Fitness Evaluation: A CNN-GRU model is trained for a certain number of epochs in order to assess each particle's position. Detection accuracy, precision, recall, and false positive rate form the basis of the evaluation metric (fitness).
- 3) Update Strategy: Each particle uses two factors to adjust its trajectory upon evaluation:
  - Personal Best (pBest): The best answer that each particle has discovered thus far.
  - Global Best (gBest): The top-performing solution found throughout the swarm. Particles are directed towards promising areas of the hyperparameter space thanks to the update process.
- 4) By doing different repetitions, particles are positioned closer to the global ideal. We have identified an ideal (or nearly ideal) set of hyperparameters when the swarm finally converges, meaning that progress has plateaued.

The CNN-GRU model is set up for a fine-tuned configuration that maximizes its performance due to the PSO-based search approach, as well as its ability to eliminate manual tuning, which is known to be inefficient. PSO vastly enhances the overall generalization of our model, reduces the false alarm rate, and enhances precision and recall by automatically identifying the best hyperparameters. It can be observed that the PSO-optimized model has significantly better performance than the baseline CNN-GRU for detection accuracy for TCP- and UDP-based DDoS attacks based on experimental results in Section 5.

**C. Explainability with XAI (SHAP):**

However, deep learning models are commonly described as black-box even when providing state-of-the-art intrusion detection capabilities. It's not enough to know, knowing that traffic has been found to be malicious in security-critical domains like DDoS detection – management must also know how the model got there. Therefore, interpretability is an important factor in adding credibility, proving accuracy, and facilitating practical solutions. To address this demand, the proposed approach uses SHapley Additive exPlanations (SHAP) to integrate Explainable Artificial Intelligence (XAI).

**Principle of SHAP:** SHAP is based on cooperative game theory, which considers every feature as a “player” contributing to the final output of the model [21]. SHAP values measure the contribution of every feature by fairly allocating the variation between the model's prediction and the average baseline prediction to every feature [22]. A negative SHAP value representing a feature supports a “Normal” classification, while a positive value increases the probability of an “Attack” classification.

**Visualization for Interpretability:**

- Waterfall Plots: The Waterfall plots illustrate for each prediction how the output probability is moved towards the “Normal” or “Attack” values by the SHAP value of each characteristic. The red bars indicate characteristics that are increasing the probability of an attack, while the blue bars indicate characteristics that are decreasing the probability. For example, the large SHAP values for dns.qry.name and dns.retransmission can also be seen in a case of DNS flood traffic and thus contributed to the classification into “Attack.”
- Global Feature Importance: With SHAP, global feature significance plots can be created by aggregating feature significance scores from the entire data set to create a global map independent of the prediction. These enable researchers and administrators to confirm that the model is dependent on important security-related properties and not

dependent on non-key feature relationships by highlighting the most significant features (e.g., tcp.flags.syn, dns.retransmission, and mqtt.msgtype).

**Practical Value in Cybersecurity:** There are many beneficial advantages of combining SHAP, including:

- **Transparency:** By enabling network managers to understand how predictions are made, automated intrusion detection systems gain more credibility.
- **Learning Validation:** SHAP verifies that the CNN-GRU model is learning the right patterns by emphasizing features at the protocol level that are already known to be related to particular types of attacks (for example, DNS queries for DNS flood attacks or irregular SYN flags for TCP floods).
- **Actionable Insights:** Findings from SHAP studies can help guide where you should monitor on the high impact, the model-derived features, if you are interested in developing protocol-specific mitigation strategies or even firewall rules.

In conclusion, the proposed CNN-GRU model framework is more appropriate for application in real-world IoT/IIoT security scenarios because of the inclusion of XAI that not only ensures high predicted accuracy but also provides interpretable, reliable, and actionable results.

## V. EXPERIMENTAL RESULT ANALYSIS

### **Performance Comparison: Before and After PSO Optimization:**

For the Edge-IIoT TCP and UDP subsets, the proposed CNN-GRU framework was tested before and after the optimization of hyperparameters by PSO. The performance of the proposed model was measured using standard classification metrics such as accuracy, precision, and recall, as well as the analysis of the confusion matrix.

#### **A. TCP Model (Decimal Scaling)**

As a result, the original TCP-based DDoS detection model had a high recall rate, but this was balanced by the high number of false positives and low precision. The performance of the model was close to perfect after PSO optimization. The confusion matrix for the original TCP model is shown in Table IV.

**TABLE IV: Confusion matrix of TCP model before PSO optimization**

|                 | Predicted<br>(0) | NormalPredicted Attack (1) |
|-----------------|------------------|----------------------------|
| True Normal (0) | 3236             | 6691                       |
| True Attack (1) | 218              | 9864                       |

After applying PSO optimization, the TCP model demonstrated a dramatic performance improvement. The updated confusion matrix is shown in Table V.

**TABLE V: Confusion matrix of TCP model after PSO optimization**

|                 | Predicted<br>(0) | NormalPredicted Attack (1) |
|-----------------|------------------|----------------------------|
| True Normal (0) | 9897             | 30                         |

|                        |   |       |
|------------------------|---|-------|
| <b>True Attack (1)</b> | 7 | 10075 |
|------------------------|---|-------|

To provide a clearer comparison, the performance metrics of the TCP model before and after PSO optimization are summarized in Table VI.

**TABLE VI: Performance metrics of TCP model before and after PSO optimization**

| <b>Model</b> | <b>Accuracy (%)</b> | <b>Precision (%)</b> | <b>Recall (%)</b> |
|--------------|---------------------|----------------------|-------------------|
| Before PSO   | 65.47               | 59.58                | 97.84             |
| After PSO    | 99.82               | 99.70                | 99.93             |

Despite the high recall value of the baseline TCP model, which indicated that it was highly sensitive to attack detection, it had a large number of misclassifications of normal traffic as attacks, resulting in a high number of false positives (6,691 instances). The performance profile of the model was well-balanced and near-perfect after applying PSO for hyperparameter tuning. The number of false negatives was reduced to only seven, and false positives were significantly reduced to just thirty. Overall, precision improved by almost 40% and accuracy by about 34%. These improvements clearly show that the TCP model was successfully transformed from an imbalanced classifier to a highly reliable detector by the optimization of PSO, making it eligible for real-world DDoS detection applications.

#### **B. UDP Model (Decimal Scaling)**

Prior to improvement, the UDP-based DDoS detection model had excellent accuracy and recall. However, PSO optimization improved its performance even further, completely removing false negatives. The confusion matrix for the baseline UDP model is shown in Table VII

**TABLE VII: Confusion matrix of UDP model before PSO optimization**

|                        | <b>Predicted Normal<br/>(0)</b> | <b>Predicted Attack (1)</b> |
|------------------------|---------------------------------|-----------------------------|
| <b>True Normal (0)</b> | 9890                            | 61                          |
| <b>True Attack (1)</b> | 6                               | 10039                       |

Following PSO optimization, the confusion matrix for the UDP model is given in Table VIII. A summary of the UDP model's performance before and after optimization is provided in Table IX.

**TABLE VIII: Confusion matrix of UDP model after PSO optimization**

|                        | <b>Predicted Normal<br/>(0)</b> | <b>Predicted Attack (1)</b> |
|------------------------|---------------------------------|-----------------------------|
| <b>True Normal (0)</b> | 9920                            | 31                          |
| <b>True Attack (1)</b> | 0                               | 10045                       |

**TABLE IX: Performance metrics of UDP model before and after PSO optimization**

| Model      | Accuracy (%) | Precision (%) | Recall (%) |
|------------|--------------|---------------|------------|
| Before PSO | 99.66        | 99.40         | 99.94      |
| After PSO  | 99.84        | 99.69         | 100.00     |

With a baseline accuracy of 99.66% and recall of 99.94%, as well as comparatively few misclassifications (only 61 false positives and 6 false negatives), the UDP model showed excellent performance even prior to tuning. The model obtained a perfect recall of 100% after PSO optimization, indicating that each assault instance in the dataset was accurately detected. Furthermore, the number of false positives was further decreased to 31, which enhanced overall reliability and precision. The total removal of false negatives will be valuable especially in the context of cyber security, since no malicious traffic could go undetected and hence increase the resilience of the defense system even though the performance improvement in this case was not as great as that obtained by the TCP model.

PSO optimization was the first to bring the most advantages to the TCP model, which evolved from a high recall but low accuracy extreme unbalanced classifier to a balanced and very accurate detector. Because packet-level anomalies tend to be more subtle and difficult to detect, this enhancement underlines the significance of hyper parameter tuning in coping with the complexities of TCP traffic. Second, while the UDP model was already robust on its own, optimization using PSO further improved its robustness. More significantly, the optimized model had 100% recall, which ensured that no cases of UDP attacks were overlooked. All these results show that the addition of PSO greatly improves the detection ability of the CNN-GRU system. The almost flawless accuracy, precision, and recall regularly obtained by the optimized models for TCP and UDP traffic validate the proposed methodology as a solid and effective alternative for DDoS detection across the IoT and IIoT architectures.

### C. Explainability Using SHAP

Explainability is indispensable for the deployment of deep learning models in cyber security, where understanding the logic behind a model's decisions is at least as important as getting high accuracy. To address the problem of the "black-box" nature of deep models, this research adopts the strategy of SHAP (SHapley Additive exPlanations), a concept rooted in cooperative game theory. SHAP assigns each input feature a contribution value that quantifies its positive or negative influence on the model's output. SHAP provides better theoretical guarantees than other interpretability techniques, works effectively with deep learning architectures, and can provide both global explanations that indicate feature importance generally and local explanations that help with explaining individual predictions. Its visual effects, like summary plots, force plots and waterfall plots, render the behavior of the complex model intuitively comprehensible.

The utility of SHAP would be best represented by the example of a waterfall plot created for a sample forecast and presented in Fig. 3. The model takes 0.5 as a baseline probability, which is just the average value before introducing features to the model. Based on this baseline, the traffic related attributes gradually alter the prediction. In this example, features such as dns.qry.qu (+0.34) and dns.qry.name (+0.12) had the highest positive effect indicating a positive influence, pushing the prediction towards the attack class. Another case of dns.qry.qu contributed -0.11, which only slightly decreased the attack's probability. Other features i.e. udp.stream, udp.port, tcp.checksum generated smaller but still relevant contributions. In addition to this summation, its final output was 0.999, which was about

certain due to the fact that the traffic instance behaved like malicious. Such step-by- step decomposition renders the model's reasoning clear and confirms its decisions are embedded in meaningful protocol-level behavior.

The explanation of SHAP gives valuable information for understanding how different aspects of normalization (Min-Max or Decimal Scaling) affect the model behaviour. In the case of the TCP Min-Max model presented in Figure 4, SHAP showed that the ultimate prediction  $f(x) = 1.0$  is effectively a function of DNS features' positive contributions (e.g., dns.qry.qu and dns.qry.name - suggesting above normal DNS query actions – typical in DNS amplification attacks. Also, lack of critical TCP flags that include FIN and ACK has its own good effect to classify the attack, meaning that connections are not quite complete or not in their true form as expected for TCP based flood attacks. These attributes together moved prediction from baseline of 0.5 to complete certainty for an attack.

So the TCP Min-Max model had the final score of  $f(x) = 1$ , which is a high confidence value for the traffic being a DDoS attack, compared to the 0.5 threshold. SHAP analysis shows that the prediction is predominantly influenced by DNS- based features, i.e. dns.qry.qu = 1 has the highest positive effect (+0.17), followed by dns.qry.name = 1 (+0.06). Those values point to very high activity in DNS queries, a clear indication of DNS-influenced flooding. Also, the missing TCP flags — tcp.connection.fin = 0 (+0.04) and tcp.flags.ack = 0 (+0.03)—indicate incomplete or unusual TCP handshakes, a signature

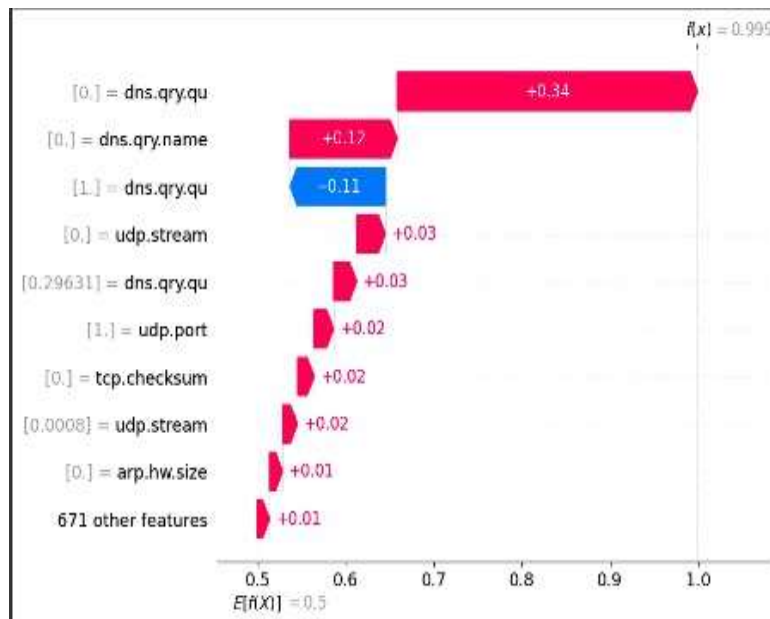


**Fig. 3: Analysis of Explainable AI Graphs for DDoS Attack Detection Models**

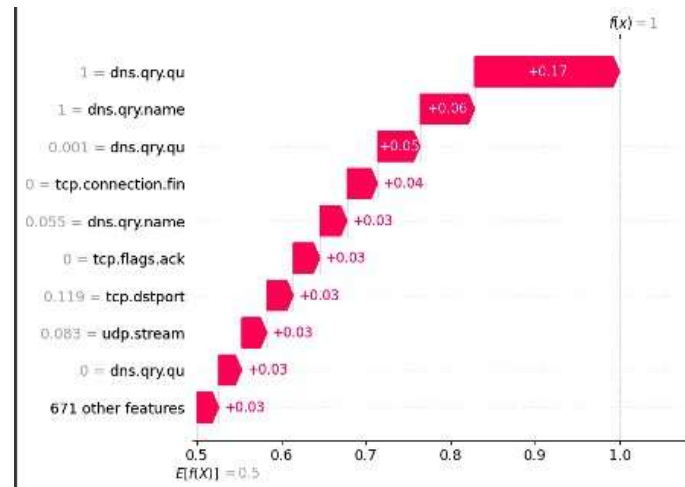


**Fig. 4: TCP Min-Max Model Explanation (Test Instance 12)**

behavior of attack traffic. Additional attributes such as low-value DNS queries, destination port indicators, and UDP stream attributes all contribute smaller but positive coefficients. These correlations highlight that this traffic has been maliciously used in DDoS attacks, as it aggregates high DNS query activity along with abnormal TCP connection behavior.

Fig. 4 shows the SHAP interpretation for the TCP Decimal Scaling model, where, in this case, the prediction is  $f(x) = 0.999$ , which means that a DDoS attack is more probable than the baseline value of 0.5. In this case, the features with missing or very low values are the key contributors, representing a major change in the interpretation of features due to decimal scaling. The positive impact is strongest from `dns.qry.qu` = [0.], which gives +0.34 to the prediction, and then comes `dns.qry.name` = [0.] (+0.12), which means that missing DNS is a critical measure of attack behavior on this scaling strategy. Further contributions come from `udp.stream` = [0.] and a low-value `dns.qry.qu` = [0.29631], corroborating both attack classification. Interestingly, a high `dns.qry.qu` = [1.] adds in the negative direction (-0.11) relative to the Min-Max model, where a great level of DNS query activity indicates an attack. It implies that at decimal scaling, the model considers missing DNS queries more suspicious than their abundant occurrence, which indicates a different learned attack signature. For this instance, the model is able to identify it as a DDoS attack with almost 100% confidence, based on only the absence of DNS and UDP activity.

To model that the UDP Min-Max model, we have presented the SHAP answer as shown in Fig.5. The final prediction using  $f(x) = 1$  shows full confidence that an instance represents a DDoS attack. This is in contrast to a baseline probability of 0.5. The `dns.qry.name` feature is the most influential, with both a moderate value (0.282) and the absence of a value playing strong roles (+0.15 each) in the classification of attack. This shows that our model has learned, in fact, that DNS query names which are missing and specific non-zero values are strong indicators of malicious behavior alike. `udp.time_delta` = 1; it adds +0.04 and indicates anomalous timing intervals between UDP packets, which are typical of UDP flood attacks. Smaller contributions from missing DNS query names and `udp.time_delta` values add further evidence to these features. Overall, the model relies on a combination of

unusual DNS query naming patterns and anomalous UDP timing patterns to make a confident determination of this traffic type as a DDoS attack.

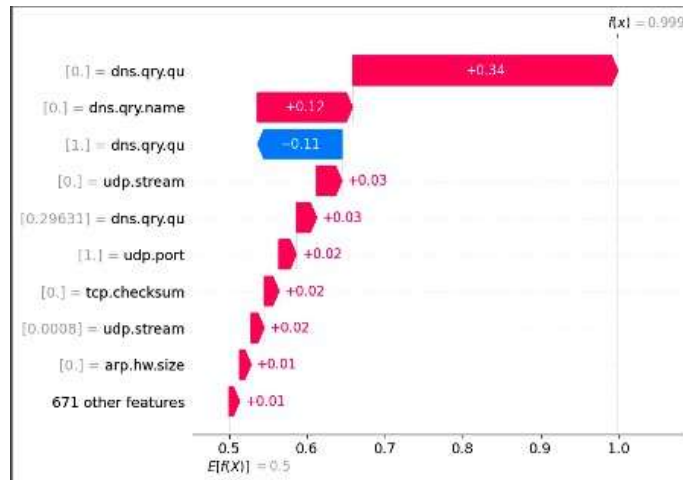


**Fig. 5: UDP Min-Max Model Explanation (Test Instance 12)**

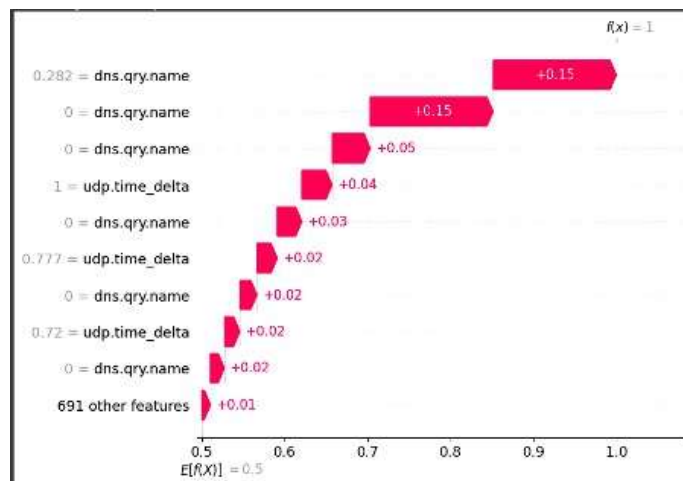


**Fig. 6: UDP Decimal Scaling Model Explanation**

The SHAP explanation for the UDP Decimal Scaling model (the final prediction  $f(x) = 1$  compared to a baseline value of 0.51) is shown in Fig. 6, indicating complete certainty of a DDoS attack. Here, the model's decision is dominated by the `dns.qry.name` feature, with the absence of a DNS query name contributing the largest positive influence (+0.22), followed by several additional zero-valued occurrences that also strongly push the prediction toward the attack class. Even a trivial non-zero value of `dns.qry.name` = 0.046 gives positive values but not to the same extent. The model adds context as mentioned above with other features, including `tcp.connection.syn` = 0, offering a slight boon, even though the attack is UDP-based. As mentioned, in general, the UDP Decimal Scaling model depends overwhelmingly on the repeated absence of a DNS query name, which shows that the model is learning to associate lack of DNS information with some specific kind of UDP-based DDoS behavior, while still incorporating smaller signals from other protocol features.

## VI. EDGE DEPLOYMENT (RASPBERRY PI)

The improved model was actually installed on a Raspberry Pi 5 edge device to see if this CNN-GRU framework is feasible after deployment and real-time evaluation. The objective of this deployment experiment was to see whether the model could provide effective detection performance even in low resource environments such as IoT and IIoT networks.

### A. Hardware and Software Setup

The deployment environment was configured using the hardware components as presented in Table X.

### B. Deployment Pipeline

A methodical pipeline was followed during the deployment process is presented in Table XI.

**TABLE X: Deployment Environment Configuration**

| Component                                        | Specification/Usage                          |
|--------------------------------------------------|----------------------------------------------|
| Raspberry Pi 5 (2.4 GHz)                         | 8 GB RAM, Quad-core ARM Cortex-A76 CPU       |
| Power Supply                                     | Official Raspberry Pi USB-C adapter (5V, 3A) |
| MicroSD Card                                     | 16 GB Class 10 (used for OS and storage)     |
| Laptop                                           | Flashing OS, remote access via SSH,          |
| transferring trained model                       |                                              |
| Network (for both Pi and laptop)                 | Mobile hotspot providing Wi-Fi connectivity  |
| Accessories                                      | MicroSD card reader for system setup         |
| Software (TensorFlow/Keras, NumPy, SHAP, Pandas) | Raspberry Pi OS, Python libraries            |
| TABLE XI: Deployment Pipeline Steps              |                                              |

| Step                  | Description                                                                                                     |
|-----------------------|-----------------------------------------------------------------------------------------------------------------|
| System Setup          | Installed Raspberry Pi OS using Raspberry Pi Imager on microSD; initial boot with monitor, keyboard, and mouse. |
| Network Configuration | Raspberry Pi and laptop connected to the same mobile hotspot for communication.                                 |
| Remote Access         | Enabled SSH, updated system software, and established a secure connection with the laptop.                      |
| Library Installation  | Installed required libraries: Pandas/Numpy for data handling, TensorFlow/Keras for model inference.             |
| Model Transfer        | Transferred trained CNN-GRU model, inference script, and test data from laptop to Raspberry Pi.                 |
| Execution             | Ran inference script on Raspberry Pi to classify traffic flows and generate predictions.                        |

For both the TCP and UDP models, the findings from the Raspberry Pi deployment were the same as those from the server environment. With its excellent precision, accuracy, and recall, the improved CNN-GRU model proved that performance was not compromised by shifting to an edge device.

Deployment success on a Raspberry Pi 5 shows that the network framework applies to IIoT and IoT networks for real-time intrusion detection. Edge deployment supports distributed threat detection closer to data sources that can reduce latency, provide scalability, and eliminate the need for central servers. This allows us to guarantee the feasibility of the proposed paradigm both philosophically and practically in real situations with limited resources.

## **VII. CONCLUSION**

Here, a hybrid CNN-GRU model for DDoS identification in IoT/IIoT networks is proposed with the help of TCP/UDP. It has been optimized through PSO for hyperparameter selection and SHAP-based explainable AI (XAI) for interpretability. Through combining CNNs' spatial learning capability with the temporal sequential model of GRUs, we were able not only to identify local abnormalities but also to identify long-range assault patterns. Experimental results demonstrated that the TCP model, which was limited to its poor baseline of 65.47% accuracy and 59.58% precision, can be optimized to a near-perfect detector with 99.82% accuracy and 99.70% precision. By comparison, the established UDP model is much better adjusted by improving its recall (100%) while producing a small number of false positives from its optimization of PSO. Such improvements obviously reflect the success of the PSO in improving the detection performance of deep learning models toward complex network traffic. Using SHAP explanations, the model does not only improve performance but also tackle the very much significant issue of interpretability. This provided clarity on feature contributions and confirmed that large protocol-level features, such as SYN flags and DNS retransmissions, were responsible for predictions. Moreover, implementing the optimized model on a Raspberry Pi 5, the results were precisely similar to server-based inference, indicating the feasibility of the model for real-time edge-based intrusion detection. Based on the results in the literature, the proposed CNN-GRU + PSO + XAI framework is a reliable, interpretable, and edge deployable solution to detect DDoS in IoT/IIoT environments. Future research will concentrate on expanding deployment across large-scale heterogeneous IoT devices, implementing federated learning for distributed training, and enhancing resilience against adversarial attacks.

## **REFERENCES**

- [1] H. Boyes, B. Hallaq, J. Cunningham, and T. Watson, "The industrial internet of things (iiot): An analysis framework," *Computers in industry*, vol. 101, pp. 1–12, 2018.
- [2] A. Amrullah, "Lightweight edge-based security system design for ddos attack mitigation in industrial iot infrastructure," *Intellithings Journal*, vol. 1, no. 1, pp. 17–24, 2024.
- [3] S. H. Mekala, Z. Baig, A. Anwar, and N. Syed, "Dos attacks, human factors, and evidence extraction for the industrial internet of things (iiot) paradigm," in *2023 38th IEEE/ACM International Conference on Automated Software Engineering Workshops (ASEW)*. IEEE, 2023, pp. 32–39.

- [4] R. S. Ahmed and T. S. Atia, "Machine learning and deep learning for distributed denial of service attack detection in software-defined networking: A review," in AIP Conference Proceedings, vol. 3211, no. 1. AIP Publishing LLC, 2025, p. 030030.
- [5] "Machine learning and deep learning for distributed denial of service attack detection in software-defined networking: A review," in AIP Conference Proceedings, vol. 3211, no. 1. AIP Publishing LLC, 2025, p. 030030.
- [6] W. G. Gadallah, H. M. Ibrahim, and N. M. Omar, "A deep learning technique to detect distributed denial of service attacks in software-defined networks," *Computers & Security*, vol. 137, p. 103588, 2024.
- [7] S. T. R. Bomidika, "Advancing ddos detection in 5g networks through machine learning and deep learning techniques," 2024.
- [8] S. Mehmood, R. Amin, J. Mustafa, M. Hussain, F. S. Alsubaei, and M. D. Zakaria, "Distributed denial of services (ddos) attack detection in sdn using optimizer-equipped cnn-mlp," *PloS one*, vol. 20, no. 1, p. e0312425, 2025.
- [9] Agrawal, V. (2017). Security with respect to MANET (mobile ad-hoc network) and IoT (internet of things). *Oriental Journal of Computer Science and Technology*, 10(1), 82-85.
- [10] Pandit, T., Raval, M., & Upadhyay, D. (2025). Multi-Domain ABSA Conversation Dataset Generation via LLMs for Real-World Evaluation and Model Comparison. *arXiv preprint arXiv:2505.24701*.
- [11] K. S. Goud and G. S. Rao, "Towards an efficient ddos attack detection in sdn: An approach with cnn-gru fusion," in 2024 Fourth International Conference on Advances in Electrical, Computing, Communication and Sustainable Technologies (ICAECT). IEEE, 2024, pp. 1–10.
- [12] S. Myneni, A. Chowdhary, D. Huang, and A. Alshamrani, "Smartdefense: A distributed deep defense against ddos attacks with edge computing," *Computer Networks*, vol. 209, p. 108874, 2022.
- [13] M. Akhi, C. Eising, and L. L. Dhirani, "Tcn-based ddos detection and mitigation in 5g healthcare-iot: A frequency monitoring and dynamic threshold approach," *IEEE Access*, 2025.
- [14] Y. Wan, K. Xu, F. Wang, and G. Xue, "Characterizing and mining traffic patterns of iot devices in edge networks," *IEEE Transactions on Network Science and Engineering*, vol. 8, no. 1, pp. 89–101, 2020.
- [15] S. S. Mahadik, P. M. Pawar, and R. Muthalagu, "Edge-hetiot defense against ddos attack using learning techniques," *Computers & Security*, vol. 132, p. 103347, 2023.
- [16] V. Hnamte, A. A. Najar, H. Nhung-Nguyen, J. Hussain, and M. N. Sugali, "Ddos attack detection and mitigation using deep neural network in sdn environment," *Computers & Security*, vol. 138, p. 103661, 2024.
- [17] M. A. Ferrag, O. Friha, D. Hamouda, L. Maglaras, and H. Janicke, "Edge-iiotset: A new comprehensive realistic cyber security dataset of iot and iiot applications for centralized and federated learning," *IEEE Access*, vol. 10, pp. 40 281–40 306, 2022.
- [18] S. Ismail, S. Dandan, and A. Qushou, "Intrusion detection in iot and iiot: Comparing lightweight machine learning techniques using ton iot, wustl-iiot- 2021, and edgeiiotset datasets," *IEEE Access*, 2025.
- [19] S. Ur Rehman, M. Khaliq, S. I. Imtiaz, A. Rasool, M. Shafiq, A. R. Javed, Z. Jalil, and A. K. Bashir, "Diddos: An approach for detection and identification of distributed denial of service (ddos) cyberattacks using gated recurrent units (gru)," *Future Generation*

- Computer Systems, vol. 118, pp. 453–466, 2021.
- [20] D. M. Rajan and D. J. Aravindhhar, "Detection and mitigation of ddos attack in sdn environment using hybrid cnn-lstm," *Migr. Lett*, vol. 20, pp. 407–419, 2023.
  - [21] Koradia, D., & Agrawal, V. (2024). Study of self-sovereign identity management system incorporating blockchain. *International Journal of Intelligent Systems and Applications in Engineering*, 12, 83-91.
  - [22] S. Gopal, D. Nanthiya, K. P. Kumar, D. H. Rajan, S. Kaviyalzini, and T. Sruthika, "Particle swarm optimization (pso) based feature selection for ddos attack detection in iot," in *2024 International Conference on Smart Electronics and Communication Systems (ISENSE)*. IEEE, 2024, pp. 1–6.
  - [23] R. R. Nuiiaa, S. A. A. A. Alsaidi, B. K. Mohammed, A. H. Alsaedi, Z. A. A. Alyasseri, S. Manickam, and M. A. Hussain, "Enhanced pso algorithm for detecting drdos attacks on ldap servers." *International Journal of Intelligent Engineering & Systems*, vol. 16, no. 5, 2023.
  - [24] C. S. Kalutharage, X. Liu, C. Chrysoulas, N. Pitropakis, and P. Papadopoulos, "Explainable ai-based ddos attack identification method for iot networks," *Computers*, vol. 12, no. 2, p. 32, 2023.
  - [25] M. Saied and S. Guirguis, "Explainable artificial intelligence for botnet detection in internet of things," *Scientific Reports*, vol. 15, no. 1, p. 7632, 2025.